

PENGEMBANGAN GAME EDUKASI LINGKUNGAN BERBASIS ANDROID

TUGAS AKHIR SKRIPSI

Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan Guna Memperoleh Gelar Sarjana Pendidikan



Oleh:

Wafda Adita Rifai

NIM. 10520241036

PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI YOGYAKARTA

2015

LEMBAR PERSETUJUAN

Tugas Akhir Skripsi dengan Judul

PENGEMBANGAN GAME EDUKASI LINGKUNGAN BERBASIS ANDROID

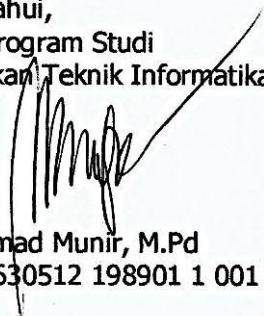
Disusun oleh:

Wafda Adita Rifai
NIM 10520241036

telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk dilaksanakan

Ujian Akhir Tugas Akhir Skripsi bagi yang bersangkutan.

Mengetahui,
Ketua Program Studi
Pendidikan Teknik Informatika,


Muhammad Munir, M.Pd
NIP. 19630512 198901 1 001

Yogyakarta, 20 Januari 2015
Disetujui,
Dosen Pembimbing,


Dr. Eko Marpanaji
NIP. 19670608 199303 1 001

HALAMAN PENGESAHAN


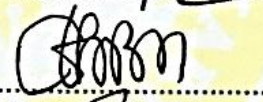

Tugas Akhir Skripsi

PENGEMBANGAN GAME EDUKASI LINGKUNGAN BERBASIS ANDROID

Disusun oleh:
Wafda Adita Rifai
NIM 10520241036

Telah dipertahankan di depan Tim Penguji Tugas Akhir Skripsi Program Studi
Pendidikan Teknik Informatika Fakultas Teknik Universitas Negeri Yogyakarta
Pada tanggal 11 Februari 2015

TIM PENGUJI

Nama/Jabatan	Tanda Tangan	Tanggal
Dr. Eko Marpanaji Ketua Penguji/Pembimbing		26/2/2015
Athika Wiji Utami, M.Pd Sekretaris		26/2/2015
Prof. Herman Dwi Surjono, Ph.D Penguji		21/2/2015

Yogyakarta, Februari 2015

Fakultas Teknik Universitas Negeri Yogyakarta

Dekan,



Dr. Moch Bruri Triyono
NIP. 19560216 198603 1 003 4

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Wafda Adita Rifai

NIM : 10520241036

Program Studi : Pendidikan Teknik Informatika

Judul TAS : Pengembangan Game Edukasi Lingkungan Berbasis
Android

menyatakan bahwa skripsi ini benar-benar karya saya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Yogyakarta, 20 Januari 2015

Yang menyatakan,



Wafda Adita Rifai
NIM . 10520241036

HALAMAN MOTTO DAN PERSEMBAHAN

A. MOTTO

“Universitas itu bukanlah suatu rumah dengan banyak kamar tapi sebuah ruang dengan banyak pintu, apapun disiplin ilmu kamu tujuannya adalah sama” Emha Ainun Najib

B. PERSEMBAHAN

Segala puji bagi Allah SWT yang senantiasa memberikan rahmat dan hidayahNya sehingga penulis mampu menyelesaikan penulisan skripsi ini.

1. Karya ini aku persembahkan untuk Ayah, Ibu dan Adik, terimakasih atas segala doa, dukungan dan pengorabanan yang telah diberikan selama ini.
2. Yang tercinta Arsita Harumsari, terima kasih atas doa dan dukungannya.
3. Tim PKM-KC Edunvi yang telah memperjuangkan Edunvi hingga PIMNAS ke 26 di Mataram dan TVRI Jogja dalam acara Kiprah Kampus yang telah mempublikasikan *Game* Edunvi.
4. Sahabat-sahabat HIMANIKA 2012 yang selalu memberikan inspirasi dan pengalaman-pengalaman berharga.
5. Teman-teman PTI-E 2010 yang pasti akan saya rindukan kebersamaanya yang tidak dapat tergantikan.

PENGEMBANGAN GAME EDUKASI LINGKUNGAN BERBASIS ANDROID

Oleh:
Wafda Adita Rifai
NIM 10520241036

ABSTRAK

Penelitian ini dilakukan dengan tujuan (1) untuk mengembangkan *Game* Edunvi pada perangkat *mobile* berbasis Android, (2) untuk mengetahui unjuk kerja dari *Game* Edunvi berdasarkan aspek *functionality*, *reliability*, *compatibility* dan *playability*, dan (3) untuk mengetahui kelayakan dari *Game* Edunvi.

Penelitian ini menggunakan metode penelitian dan pengembangan (R&D). Langkah yang dilakukan menggunakan V-model yang meliputi 1) Analisis Kebutuhan, 2) Analisis Spesifikasi, 3) Desain, 4) Implementasi, 5) *Unit Testing*, 6) *Integration Testing*, 7) *System Testing* dan 8) *Acceptance Testing*. Teknik pengumpulan data yang dilakukan adalah observasi dan kuesioner. Subjek penelitian ini adalah Aplikasi *Game* Edunvi dan 20 orang pengguna aktif Android.

Berdasarkan prosedur pengujian produk yang dilakukan diperoleh hasil : 1) perancangan *Game* Edunvi telah melalui beberapa proses pengembangan perangkat lunak dengan menggunakan V-Model, 2) hasil pengujian aplikasi diperoleh nilai aspek *functionality* dengan presentase 100% (sangat baik), pengujian aspek *reliability* dengan presentase sebesar 100% (sangat baik), pengujian aspek *compability* dengan presentase sebesar 100% (sangat baik), pengujian *playability* dengan presentase sebesar 82% (sangat layak) dan nilai Alpha Cronbach sebesar 0.923, dan 3) hasil pengujian rata-rata menunjukkan presentase sebesar 96% (sangat layak).

Kata kunci : *game edukasi, lingkungan, android, V-Model, anak-anak, functionality, reliability, compability, playability*.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan Kehadirat Tuhan Yang Maha Esa atas kuasa dan limpahan rahmat-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Pendidikan Teknik di Fakultas Teknik Universitas Negeri Yogyakarta.

Penelitian ini memberikan banyak pelajaran-pelajaran mengenai apa yang menjadi fokus materi yang penulis kembangkan yaitu Pengembangan Game Edukasi Lingkungan Berbasis Android. Selama penyusunan skripsi ini, penulis tidak terlepas dari bantuan, bimbingan dan peran dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak berikut :

1. Bapak Dr. Eko Marpanaji selaku Dosen Pembimbing TAS yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan Tugas Akhir Skripsi.
2. Ibu Dessy Irmawati, M.T dan Bapak Suparman, M.Pd selaku validator instrument penelitian TAS yang memberikan saran dan perbaikan sehingga penelitian TAS dapat terlaksana sesuai dengan tujuan.
3. Bapak Ponco Wali, Rahadian Pradipta, dan Ibu Dr Insih Wilujeng, M.Pd selaku pakar yang telah memberikan masukan dan perbaikan terhadap perangkat lunak yang dikembangkan.
4. Bapak Muhammad Munir, M.Pd selaku Ketua Jurusan Pendidikan Teknik Elektronika dan Ibu Dr. Ratna Wardani selaku Ketua Program Studi Pendidikan Teknik Informatika yang telah memberikan kemudahan dalam penyelesaian Tugas Akhir Skripsi ini.

5. Bapak Dr. Mochamad Bruri Triyono, M.Pd selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta yang telah memberikan izin penelitian serta segala kemudahan yang diberikan.
6. Semua pihak yang telah banyak mendukung dan membantu pelaksanaan penelitian ini.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini, untuk itu penulis mengharapkan kritik dan saran yang sifatnya membangun agar dalam pengembangannya kedepan lebih baik. Semoga laporan penelitian skripsi memberikan manfaat bagi pembaca dan masyarakat luas.

Yogyakarta, 20 Januari 2015

Penulis,

Wafda Adita Rifai
10520241036

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
HALAMAN PENGESAHAN	iii
SURAT PERNYATAAN	iv
HALAMAN MOTTO DAN PERSEMBAHAN	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Identifikasi Masalah	4
C. Pembatasan Masalah	4
D. Perumusan Masalah.....	4
E. Tujuan Penelitian.....	5
F. Manfaat Penelitian	5
1. Manfaat Teoritis	5
2. Manfaat Praktis	5
BAB II KAJIAN PUSTAKA	6
A. Kajian Teori	6
1. <i>Game</i>	6
2. <i>Game</i> Edukasi	7
3. Android.....	8
4. <i>Unified Model Language</i>	10
5. Pengujian Perangkat Lunak	18
6. <i>Construct 2</i>	25

7. <i>Test Object</i>	27
B. Hasil Penelitian yang Relevan	28
C. Kerangka Pikir	29
BAB III METODE PENELITIAN	31
A. Desain Penelitian	31
B. Metode Pengembangan.....	32
1. Analisis Kebutuhan	32
2. Analisis Spesifikasi	32
3. Desain	33
4. Implementasi	34
C. Metode Pengujian.....	34
D. Variabel Penelitian	35
E. Subjek Penelitian	36
F. Tempat dan Waktu Pelaksanaan	36
G. Teknik Pengumpulan Data.....	36
1. Observasi.....	36
2. Kuisioner.....	36
H. Instrumen Penelitian.....	37
1. <i>Functionality</i>	37
2. <i>Reliability</i>	37
3. <i>Compatibility</i>	37
4. <i>Playability</i>	38
I. Analisis Data	40
1. Analisis <i>Functionality, Reliability</i> dan <i>Compatibity</i>	40
2. Analisis Aspek <i>Playability</i>	41
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	43
A. Pengembangan Perangkat Lunak	43
1. Analisis Kebutuhan	43
2. Analisis Spesifikasi	45
3. Desain	47
4. Implementasi	67
B. Pengujian Perangkat Lunak	67

1. <i>Unit Testing</i>	67
2. <i>Integration Testing</i>	77
3. <i>System Testing</i>	79
4. <i>Acceptance Testing</i>	82
C. Revisi Produk	86
D. Hasil Akhir Produk	87
BAB V SIMPULAN DAN SARAN	88
A. Kesimpulan	88
B. Saran	89
DAFTAR PUSTAKA	90
LAMPIRAN	93

DAFTAR GAMBAR

	Halaman
Gambar 1. Klasifikasi Diagram UML	11
Gambar 2. Notasi <i>Class Diagram</i>	15
Gambar 3. V-Model	20
Gambar 4. Tahap Pengujian	22
Gambar 5. Halaman Website <i>Construct 2</i>	26
Gambar 6. Halaman Website <i>TestObject</i>	27
Gambar 7. Bagan Kerangka Pikir Penelitian	30
Gambar 8. Diagram Alur Penelitian	32
Gambar 9. Distribusi Sistem Operasi Android	45
Gambar 10. <i>Use Case</i> Edunvi	52
Gambar 11. <i>Diagram Sequence</i> memilih level	56
Gambar 12. <i>Diagram Sequence Reset</i> Skor	56
Gambar 13. <i>Diagram Sequence</i> Memainkan <i>Game</i>	57
Gambar 14. <i>Class Diagram</i> Edunvi	58
Gambar 15. <i>Activity Diagram</i> Memilih Level	59
Gambar 16. <i>Activity Diagram</i> Memainkan Permainan	60
Gambar 17. <i>Activity Diagram</i> Reset Level	61
Gambar 18. Rancangan Tampilan Menu	62
Gambar 19. Rancangan Tampilan <i>Level</i>	63
Gambar 20. Rancangan Tampilan <i>Level 1</i>	63
Gambar 21. Rancangan Tampilan <i>Level 2</i>	63
Gambar 22. Rancangan Tampilan <i>Level 3</i>	64
Gambar 23. Rancangan Tampilan <i>Help</i>	64
Gambar 24. Tampilan Menu Edunvi	65
Gambar 25. Tampilan Pilih Level Edunvi	65
Gambar 26. Tampilan <i>Level 1</i> Edunvi	65
Gambar 27. Tampilan <i>Level 2</i> Edunvi	66
Gambar 28. Tampilan <i>Level 3</i> Edunvi	66
Gambar 29. Tampilan <i>Help</i> Edunvi	66
Gambar 30. Implementasi <i>Event</i>	67
Gambar 31. <i>Flowchart</i> Edunvi	69
Gambar 32. Notasi <i>Flowgraph</i>	70
Gambar 33. <i>Flowgraph</i> Edunvi	71
Gambar 34. Jumlah perangkat yang mendukung Edunvi menurut <i>Playstore</i>	80
Gambar 35. Hasil Penghitungan <i>Reability</i> dengan <i>Alpha Cronbach</i>	85

DAFTAR TABEL

	Halaman
Tabel 1. Deskripsi Diagram UML.....	11
Tabel 2. Notasi <i>Use Case</i>	13
Tabel 3. Notasi <i>Sequence Diagram</i>	16
Tabel 4. Notasi <i>Activity Diagram</i>	17
Tabel 5. Teknik Pengujian Perangkat Lunak	19
Tabel 6. Metode Pengujian Edunvi.....	34
Tabel 7. Kuisisioner <i>Playability</i>	39
Tabel 8. Skala Penilaian Media (konversi nilai).....	41
Tabel 9. Skala Penilaian Likert.....	42
Tabel 10. Spesifikasi <i>Hardware</i> untuk Pengembangan.....	45
Tabel 11. Spesifikasi Edunvi.....	46
Tabel 12. Deskripsi Level pada Edunvi	48
Tabel 13. Definisi Aktor	49
Tabel 14. Definisi <i>Use Case</i>	50
Tabel 15. Skenario Membuka Aplikasi	52
Tabel 16. Skenario Membuka <i>Help</i>	53
Tabel 17. Skenario Membuka <i>Play</i>	53
Tabel 18. Skenario Memilih Level	53
Tabel 19. Skenario <i>Reset Level</i>	54
Tabel 20. Skenario Memainkan <i>Game</i>	54
Tabel 21. Skenario Data <i>Score</i>	55
Tabel 22. Skenario Keluar Aplikasi	55
Tabel 23. Hubungan Kompleksitas dan Resiko	73
Tabel 24. Uji <i>Test Case</i>	75
Tabel 25. Hasil <i>Blackbox Test</i>	77
Tabel 26. Hasil pengujian <i>Compatibility</i>	79
Tabel 27. Hasil <i>Stress Testing</i>	81
Tabel 28. Hasil Pengujian Ahli	82
Tabel 29. Saran dari Ahli	83
Tabel 30. Hasil Uji <i>Playability</i>	84
Tabel 31. Hasil pengujian Game Edunvi	86
Tabel 32. Daftar Perbaikan <i>Game</i> Edunvi	87
Tabel 33. Hasil Publikasi <i>Game</i> Edunvi.....	87

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. <i>Screenshoot Event</i> Edunvi	94
Lampiran 2. <i>Quality Report</i> Edunvi	98
Lampiran 3. Edunvi Pada Perangkat Android	109
Lampiran 4. Hasil Pengujian <i>Playability</i>	111
Lampiran 5. Validasi Instrumen	112
Lampiran 6. Hasil <i>Expert Judgement</i>	116
Lampiran 7. Kartu Bimbingan Skripsi	125
Lampiran 8. Surat Keterangan Pembimbing	126

BAB I

PENDAHULUAN

A. Latar Belakang

Di Indonesia jumlah sampah mengalami peningkatan seiring dengan pertumbuhan penduduk. Kementerian Lingkungan Hidup mencatat pada tahun 2012 rata-rata penduduk Indonesia menghasilkan sampah sekitar 2 kg per orang per hari. Berdasarkan perhitungan tersebut dapat diperkirakan berapa banyak volume sampah yang dihasilkan oleh suatu kota setiap hari dengan mengalikan jumlah penduduknya dengan 2 kg per orang per hari.

Kementerian Lingkungan Hidup (2012) menyatakan bahwa volume sampah dalam tiga tahun terakhir menunjukkan kenaikan secara signifikan. Volume sampah pada tahun 2010 ada 200.000 ton/hari dan pada tahun 2012 ada 490.000 ton per hari atau total 178.850.000 ton setahun. Dari total sampah tersebut lebih dari 50% adalah sampah rumah tangga.

Sampah rumah tangga yang jumlahnya lebih dari 50% total sampah ternyata belum ditangani dengan baik. Baru sekitar 24,5% sampah rumah tangga di Indonesia yang ditangani dengan metode yang benar yaitu diangkut oleh petugas kebersihan dan dikomposkan. Sisanya (75,5%) belum ditangani dengan baik. Fakta itu ditunjukkan oleh data RISKESDAS 2010 yang menyatakan bahwa rumah tangga di Indonesia umumnya menerapkan 6 metode penanganan sampah, yaitu: 1) diangkut oleh petugas kebersihan (23,4%), 2) dikubur dalam tanah (4,2%), 3)

dikomposkan (1,1%), 4) dibakar (52,1%), 5) dibuang di selokan/sungai/laut (10,2%) dan 6) dibuang sembarangan (9%) (Kantor Utusan Khusus Presiden RI untuk MDGs, 2012).

Fakta penanganan sampah tersebut di atas juga menunjukkan perilaku masyarakat yang belum mempedulikan sampah rumah tangganya terlebih terhadap lingkungan. Hal tercermin dari budaya masyarakat kita yang masih sangat gemar untuk membuang sampah secara sembarangan. Oleh karena itu harus ada upaya pencegahan dan penanaman sikap peduli terhadap lingkungan sejak dini.

Sementara yang terjadi saat ini adalah media pendidikan tentang pengelolaan sampah yang ada masih kurang jumlahnya dan tidak menarik. Menurut Daryanto (2014) menyebutkan bahwa materi dan metode pelaksanaan Pendidikan Lingkungan Hidup yang selama ini digunakan dirasakan belum memadai sehingga pemahaman kelompok sasaran mengenai pelestarian lingkungan hidup menjadi tidak utuh. Di samping itu, materi dan metode pelaksanaan Pendidikan Lingkungan Hidup yang tidak aplikatif kurang mendukung penyelesaian permasalahan lingkungan hidup yang dihadapi di daerah masing-masing. Mengingat berkembangnya teknologi saat ini yang cukup pesat, peneliti mencoba memecahkan persoalan ini melalui rekayasa perangkat lunak yang dirasa cukup murah, populer dan digemari anak-anak.

Dalam kurun waktu 10 tahun terakhir, perkembangan teknologi informasi terutama dalam hal perangkat lunak berlangsung sangat cepat. Hal ini ditandai dengan munculnya berbagai perangkat yang sudah

mendukung komputasi dengan segala fiturnya. Salah satunya perangkat komputasi bergerak melalui *smartphone*, pengguna *smartphone* di Indonesia diprediksi mencapai hingga 82 juta pada tahun 2014. (Teknojurnal,2011). Perkembangan teknologi yang canggih ini juga harus didukung dengan sistem operasi yang terbaru. Salah satunya adalah sistem operasi Android. Pada tahun 2012, pengguna Android di Indonesia meningkat 22%. Perangkat berbasis *operating system* (OS) Android, kini mulai bermunculan dan makin beragam di pasar, hal ini dikarenakan Android merupakan platform terbuka, sehingga bisa dijalankan di berbagai perangkat *mobile & Internet devices* (MID). Selain itu, Android memiliki beberapa kelebihan diantaranya adalah banyaknya vendor yang mengadopsi sistem operasi Android pada perangkatnya, *operating system* yang bersifat *open source*, tersedia market / pasar aplikasi. Keunggulan lainnya adalah kemudahan, interaktifitas dan *user experience* yang disajikan Android melalui aplikasinya, salah satunya adalah jenis *game*. Seperti kita ketahui, mayoritas anak-anak sangat menyukai *game*. Berdasarkan data dari Appbrain (2015) menunjukkan bahwa jumlah *game* pada *Google Play* yaitu *Game Puzzle* sejumlah 59.283, *Game Casual* 51.458, *Game Arcade* 47.283, *Game Action* 17.853, *Game Educational* 14.180, *Game Adventure* 10.018. Hal ini menunjukkan bahwa jenis *game* edukasi masih kurang dibandingkan dengan *genre* game lain.

Ditinjau dari fakta-fakta di atas, maka peneliti berinovasi untuk menyediakan fitur pembelajaran lingkungan, terutama dalam kaitan pengolahan sampah untuk anak-anak. Aplikasi yang dibuat ini nantinya

akan dikemas dalam bentuk game interaktif berbasis Android. Game yang diciptakan ini harapannya akan memberikan pengetahuan dan menciptakan kesadaran kepada anak-anak akan pentingnya menjaga lingkungan. Pola hidup cinta lingkungan yang ditanamkan ini diawali dari hal-hal kecil dalam kehidupan sehari-hari yaitu membuang sampah secara benar.

B. Identifikasi Masalah

Dari latar belakang diatas terdapat beberapa masalah yang terlihat seperti :

1. Kurang pedulinya masyarakat akan pentingnya kelestarian lingkungan.
2. Media edukasi lingkungan yang ada dirasa belum memadai.
3. Jumlah *game* edukasi pada *Playstore* masih kurang dibandingkan *genre game* yang lain.

C. Pembatasan Masalah

Dengan mempertimbangkan permasalahan yang akan dihadapi, penulis membatasi permasalahan pada pengembangan game edukasi lingkungan berbasis Android.

D. Perumusan Masalah

Berdasarkan identifikasi masalah dapat dirumuskan sebagai berikut :

1. Bagaimana mengembangkan Game Edunvi pada perangkat berbasis Android?

2. Bagaimana unjuk kerja Game Edunvi pada perangkat berbasis Android?
3. Bagaimana tingkat kelayakan Game Edunvi pada perangkat berbasis Android?

E. Tujuan Penelitian

Tujuan penelitian ini adalah untuk :

1. Mengembangkan Game Edunvi pada perangkat berbasis Android.
2. Mengetahui unjuk kerja Game Edunvi pada perangkat berbasis Android.
3. Mengetahui kelayakan dari Game Edunvi pada perangkat berbasis Android.

F. Manfaat Penelitian

1. Manfaat Teoritis

Hasil penelitian ini dapat dijadikan referensi untuk pengembangan game berbasis Android.

2. Manfaat Praktis

- a. Mengetahui kelayakan dari kualitas Game Edunvi.
- b. Game Edunvi yang telah lulus uji kelayakan dapat dijadikan media edukasi tentang lingkungan pada anak-anak.

BAB II

KAJIAN PUSTAKA

A. Kajian Teori

1. *Game*

Menurut Clark (2006) *game* adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan dibatasi oleh konteks tertentu. Menurut Arief S. Sadiman (2010) *game* adalah kompetisi antara para pemain yang berinteraksi satu sama lain dengan menggunakan aturan-aturan tertentu untuk mencapai tujuan-tujuan tertentu pula. Dalam sebuah permainan harus ada kompetisi agar pemain terangsang untuk terus bermain, kompetisi tersebut dapat berwujud menang dan kalah. Pemain harus bisa menemukan strategi atau cara untuk memecahkan masalah sehingga dapat memenangkan game tersebut.

Game adalah permainan yang menggunakan media elektronik, merupakan sebuah hiburan berbentuk multimedia yang di buat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasan batin. Bermain *game* merupakan salah satu sarana pembelajaran. *Game* lebih sering dimainkan oleh anak-anak, akan tetapi pada zaman sekarang orang dewasa juga suka bermain game dan mengikuti perkembangan game-game yang ada sekarang. Jenis *game* sangatlah tergantung dari perkembangan zaman. Jika dilihat dari grafis yang digunakan dalam aplikasi permainan, maka aplikasi permainan dapat

digolongkan menjadi dua jenis, yaitu aplikasi permainan 2D (dua dimensi) dan 3D (tiga dimensi).

Game jika dilihat dari cara memainkannya memiliki berbagai *genre* atau aliran diantaranya : *First Person Shooter* (permainan aksi dengan sudut pandang orang pertama), *Role Play Games* (memerankan tokoh), *Arcade* (ketangkasan), *Adventure* (pertualangan), *Simulation* (simulasi) dan lain sebagainya. Salah satu *genre* yang sedang menjadi tren adalah *game arcade*. *Game arcade* adalah jenis permainan yang mengandalkan ketangkasan tangan pemainnya dalam melakukan kontrol. Ciri *game arcade* yang umum yakni biasanya memiliki konsep dan desain yang simpel dan tingkat kesulitan yang bertambah di setiap level.

2. *Game* Edukasi

Game edukasi adalah permainan yang dirancang atau dibuat untuk merangsang daya pikir termasuk meningkatkan konsentrasi dan memecahkan masalah (Handriyantini, 2009). *Game* Edukasi adalah salah satu jenis media yang digunakan untuk memberikan pengajaran, menambah pengetahuan penggunaannya melalui suatu media unik dan menarik. *Game* edukasi dibuat dengan tujuan spesifik sebagai alat pendidikan, untuk belajar mengenal warna, mengenal huruf dan angka, matematika, sampai belajar bahasa asing.

Game dengan tujuan edukasi seperti ini dapat digunakan sebagai salah satu media edukasi yang memiliki pola pembelajaran *learning by doing*. Berdasarkan pola yang dimiliki oleh game tersebut, pemain dituntut untuk belajar sehingga dapat menyelesaikan permasalahan yang

ada. Status *game*, instruksi, dan *tools* yang disediakan oleh *game* akan membimbing pemain secara aktif untuk menggali informasi sehingga dapat memperkaya pengetahuan dan strategi saat bermain.

Berdasarkan hasil penelitian penelitian sebelumnya, tidak diragukan lagi bahwa game edukasi dapat menunjang proses pendidikan (Marsh, dkk, 2005; Clark, 2006). *Game* edukasi unggul dalam beberapa aspek jika dibandingkan dengan metode pembelajaran konvensional. Salah satu keunggulan yang signifikan adalah adanya animasi yang dapat meningkatkan daya ingat sehingga anak dapat menyimpan materi pelajaran dalam waktu yang lebih lama dibandingkan dengan metode pengajaran konvensional (Clark, 2006).

Game edukasi adalah permainan yang digunakan dalam proses pembelajaran dan dalam permainan tersebut mengandung unsur mendidik atau nilai-nilai pendidikan. *Game* edukasi yang akan dikembangkan bergenre *Arcade*, dimana melatih ketangkasan pengguna dalam menggolongkan sampah. Serta memiliki esensi simulasi didalamnya, sehingga membiasakan pengguna untuk membuang sampah pada tempatnya dan jenisnya.

3. Android

Pengembangan aplikasi permainan Edunvi ini berbasis pada sistem operasi Android. Terdapat berbagai macam definisi Android oleh beberapa ahli, salah satunya Sfaat (2012) menyatakan bahwa Android adalah sebuah sistem operasi perangkat mobile berbasis linux. Sedangkan menurut J.F. DiMarzio (2008), Android merupakan sebuah sistem operasi

berbasis Java yang beroperasi pada kernel Linux 2.6. Android bukanlah sebuah bahasa pemrograman tetapi Android merupakan sebuah lingkungan untuk menjalankan aplikasi.

Android menyediakan platform terbuka/*open source* bagi para pengembang sehingga menjadikan sistem operasi ini sangat digemari di pasaran. Sebagian besar vendor *smartphone* yang diproduksi adalah berbasis Android. Hal ini juga yang menjadikan banyak pengembang mulai mengembangkan aplikasi berbasis Android.

Versi android yang dijadikan *platform* pengembangan *game* Edunvi adalah Android versi 2.3 keatas yaitu :

a. *Gingerbread* (2.3)

Perubahan utama di versi ini termasuk *update* UI, peningkatan fitur *soft keyboard & copy/ paste, power management*, dan *Near Field Communication*.

b. *Honeycomb* (3.0-3.2)

Android *Honeycomb* dikhususkan untuk perangkat tablet. Fitur yang disediakan memang disesuaikan dengan kebutuhan pada perangkat layar lebar. Salah satu perubahan yang cukup besar adalah pada tampilan yang lebih profesional

c. *Ice Cream Sandwich* (3.2-4.0)

Android *Ice Cream Sandwich* atau biasa disingkat dengan ICS adalah android pertama yang memiliki fitur membuka kunci dengan pengenalan wajah. Perkembangan versi ini sangat terasa pada tampilan *interface* bersih dan *smooth*

d. *Jelly Bean* (4.1-4.3)

Keunggulan Android *Jelly Bean* dibanding seri terdahulu antara lain: Sistem Keamanan (Sekuritas), *Dual Boot*, *File Manager*, *Keyboard Virtual*.

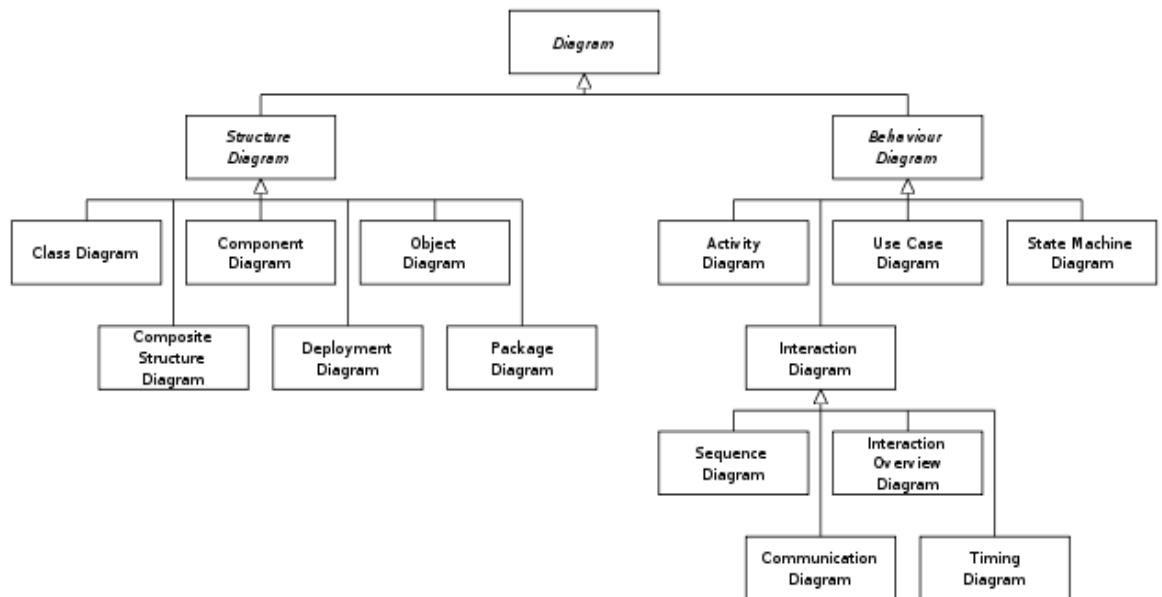
Sistem operasi Android *Jelly Bean* ini akan lebih ringan disertai kinerja dalam mengakses aplikasi pada *smartphone*/komputer lebih cepat, terdapat aplikasi khusus penghemat daya baterai yang langsung *built-in* tanpa harus melakukan *download* di Android *market*, dan kelebihan lainnya android versi di atas 4.1 juga sangat dimungkinkan untuk menjalankan perangkat lunak yang dikembangkan.

4. *Unified Model Language*

Menurut Fowler(2004) UML adalah kumpulan notasi grafis yang membantu dalam mengembangkan dan merancang sistem perangkat lunak, khususnya sistem perangkat lunak yang dibangun dengan *object oriented*. Menurut Nugroho (2010), "UML (*Unified Modeling Language*) adalah 'bahasa' pemodelan untuk sistem atau perangkat lunak yang berparadigma 'berorientasi objek". Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami".

Berdasarkan pendapat yang dikemukakan di atas dapat ditarik kesimpulan bahwa "*Unified Modelling Language* (UML) adalah sebuah bahasa pemodelan yang merepresentasikan dan memvisualkan sebuah sistem pengembangan perangkat lunak berbasis *object oriented*. UML

memberikan 13 jenis diagram yang digunakan untuk merepresantasikan bagian penting dari sebuah rancangan perangkat lunak. Penggunaan jenis diagram didasarkan pada kebutuhan dan karakteristik sebuah perangkat lunak. Berikut ini adalah jenis-jenis diagram beserta fungsinya :



Gambar 1. Klasifikasi Diagram UML

Tabel 1. Deskripsi Diagram UML

Diagram	Kegunaan	Learning Priority
<i>Activity</i>	Menggambarkan proses aplikasi tingkat tinggi, termasuk aliran data, atau untuk model logika yang kompleks dalam sistem	<i>High</i>
<i>Class</i>	Menunjukkan sekumpulan elemen model statis seperti kelas dan jenis, isinya, dan hubungan mereka.	<i>High</i>
<i>Communication</i>	Menunjukkan contoh dari kelas, hubungan antar kelas, dan alur pesan antara mereka	<i>Low</i>
<i>Component</i>	Menggambarkan komponen yang membentuk sebuah aplikasi	<i>Medium</i>
<i>Composite structure</i>	Menggambarkan struktur internal dari sebuah pengklasifikasi	<i>Low</i>

Diagram	Kegunaan	Learning Priority
<i>Deployment</i>	Pemindahan artifak ke node	<i>Medium</i>
<i>Interaction overview</i>	Sebuah varian dari diagram aktivitas yang mengontrol aliran dalam proses sistem	<i>Low</i>
<i>Object</i>	Menggambarkan objek dan relasi mereka di sebuah titik waktu tertentu	<i>Low</i>
<i>Package</i>	Menunjukkan bagaimana elemen model akan disusun dalam paket maupun dependensi antara bentuk paket	<i>Low</i>
<i>Sequence</i>	Interaksi antar obyek, penekanan pada <i>sequence</i>	<i>High</i>
<i>State machine</i>	Menjelaskan lingkungan sebuah objek atau interaksi di dalamnya, maupun transisi antara lingkungan	<i>Medium</i>
<i>Timing</i>	Menggambarkan perubahan keadaan dari waktu ke waktu	<i>Low</i>
<i>Use case</i>	Bagaimana pengguna berinteraksi dengan sistem	<i>Medium</i>

Dalam pengembangan aplikasi permainan Edunvi, peneliti hanya akan menggunakan 4 macam diagram UML yaitu *Use Case Diagram*, *Class Diagram*, *Sequence Diagram* dan *Activity Diagram*. Pemilihan diagram ini didasarkan pada tingkat *Learning Priority* yang tinggi serta dirasa cukup mewakili dari seluruh segmentasi pada diagram UML. Notasi diagram-diagram tersebut adalah sebagai berikut :



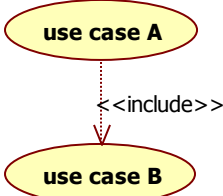
a. *Use Case Diagram*






Menurut John Satzinger (2010) *use case* adalah sebuah kegiatan yang dilakukan oleh sistem yang biasanya menanggapi permintaan dari pengguna sistem. Menurut Henderi (2008) *use case diagram* secara grafis menggambarkan interaksi antara sistem, sistem eksternal dan pengguna. Dengan kata lain *use case diagram* secara grafis mendeskripsikan siapa

yang akan menggunakan sistem dan dalam cara apa pengguna mengharapkan interaksi dengan sistem itu.

Setiap *use case* nantinya akan didefinisikan lebih lanjut melalui sebuah skenario. Notasi *use case* adalah sebagai berikut :

Tabel 2. Notasi *Use Case*

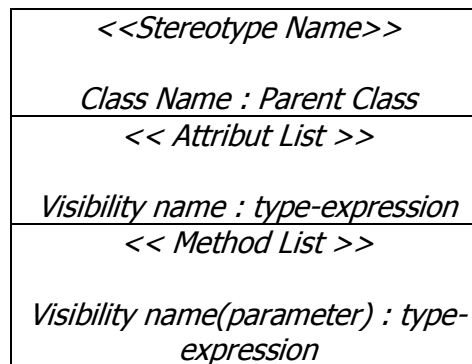
Simbol	Nama	Kegunaan
	<i>Actor</i>	Aktor adalah segala hal diluar sistem (bisa manusia, sistem, atau perangkat) yang akan menggunakan sistem tersebut untuk melakukan sesuatu.
	<i>Dependency</i>	<i>Dependency</i> merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain. Elemen yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah. Terdapat dua tipe yaitu <i>include</i> dan <i>extend</i> .
 <pre> graph TD A([use case A]) -.-> <<include>> B([use case B]) </pre>	<i>Include</i>	Menunjukkan bahwa suatu bagian dari elemen (yang ada digaris tanpa panah) memicu eksekusi bagian dari elemen lain (yang ada di garis dengan panah). <i>Use case A</i> dapat berjalan jika <i>use case B</i> sudah dijalankan minimal satu kali
 <pre> graph TD A([use case A]) -.-> <<extend>> B([use case B]) </pre>	<i>Extend</i>	Menunjukkan bahwa suatu bagian dari elemen di garis tanpa panah bisa disisipkan kedalam elemen yang ada di garis dengan panah. <i>Use case A</i> memanggil <i>use case B</i> pada kondisi tertentu

Simbol	Nama	Kegunaan
	<i>Generalization</i>	Menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik. Dengan <i>generalization</i> , <i>class</i> yang lebih spesifik (<i>subclass</i>) akan menurunkan atribut dan operasi dari <i>class</i> yang lebih umum (<i>superclass</i>)
	<i>Association</i>	Mengidentifikasi interaksi antara setiap aktor tertentu dengan setiap use case tertentu. Digambarkan sebagai garis antara aktor terhadap <i>use case</i> yang bersangkutan.
	<i>System</i>	Menyatakan batasan sistem dalam relasi dengan aktor-aktor yang menggunakannya (di luar sistem) dan fitur-fitur yang harus disediakan (dalam sistem). Sistem disertai label yang menyebutkan nama dari sistem.
	<i>Use case</i>	Mengidentifikasi fitur kunci dari sistem. Tanpa fitur ini, sistem tidak akan memenuhi permintaan user/aktor. Setiap <i>use case</i> mengekspresikan tujuan dari sistem yang harus dicapai dan diberi nama sesuai dengan tujuannya.
	<i>Interaction</i>	Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek. Bisa sanya <i>interaction</i> ini dilengkapi juga dengan teks bernama <i>operation signature</i> yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

b. Class Diagram

John Satzinger (2010) menyatakan bahwa *Class Diagram* adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun. Menurut Henderi (2008), *class diagram* menggambarkan struktur objek sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut.

Notasi *Class Diagram* adalah sebagai berikut :




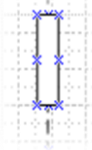



Gambar 2. Notasi *Class Diagram*

c. Sequence Diagram

Menurut John Satzinger (2010) *sequence diagram* adalah diagram yang digunakan untuk mendefinisikan input dan output serta urutan interaksi antara pengguna dan sistem untuk sebuah *use case*. Menurut Henderi (2008), *sequence diagram* secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah *use case* atau operasi. Notasi *sequence diagram* adalah sebagai berikut :

Tabel 3. Notasi *Sequence Diagram*

Simbol	Nama	Kegunaan
	<i>Object</i>	Merupakan instance dari sebuah <i>class</i> dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah <i>class</i> (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma.
	<i>Actor</i>	Aktor adalah segala hal diluar sistem (bisa manusia, sistem, atau perangkat) yang akan menggunakan sistem tersebut untuk melakukan sesuatu
	<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.
	<i>Activation</i>	<i>activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
	<i>Message</i>	<i>message</i> digambarkan dengan anak panah horizontal antara <i>Activation</i> . <i>Message</i> mengindikasikan komunikasi antara object-object.

d. Activity Diagram

John Satzinger (2010) menyatakan bahwa "*Activity Diagram* adalah sebuah alur kerja yang menjelaskan berbagai kegiatan pengguna (atau sistem), orang yang melakukan aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut". Sedangkan menurut Henderi (2008), *activity diagram* secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas *use case*. *Activity diagram* dapat juga digunakan untuk

memodelkan aksi yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari aksi tersebut.

Notasi pada *Activity Diagram* adalah :

Tabel 4. Notasi *Activity Diagram*

Simbol	Nama	Kegunaan
	<i>Start Point</i>	Menunjukkan dimana aliran kerja dimulai
	<i>End Point</i>	Menunjukkan dimana aliran kerja berakhir
	<i>Activities</i>	Menunjukkan kegiatan dalam aliran kerja
	<i>Dicision</i>	Menunujukan dimana sebuah keputusan perlu diambil dalam aliran kerja
	<i>Join</i>	Menunjukkan percabangan pada aliran kerja
	<i>Fork</i>	Menunjukkan penggabungan dalam aliran kerja

5. Pengujian Perangkat Lunak

Pengujian perangkat lunak menurut IEEE adalah proses mengoperasikan sistem atau komponen dalam kondisi tertentu, mengamati atau merekam hasil, dan membuat evaluasi terhadap beberapa aspek dari sistem atau komponen. Sedangkan menurut Glen Myres(2004), pengujian adalah proses menjalankan program dengan maksud menemukan kesalahan. Dari berbagai pendapat ahli diatas dapat disimpulkan bahwa pengujian perangkat lunak adalah proses identifikasi pada perangkat lunak untuk memperoleh informasi terkait kualitas dari sebuah perangkat lunak yang sedang diuji.

Pressman menyebutkan bahwa pengujian perangkat lunak sebenarnya merupakan sebuah proses verifikasi dan validasi. Verifikasi adalah tahapan dari rekayasa perangkat lunak untuk memastikan produk yang dihasilkan dari aktivitas pengembangan sesuai dengan spesifikasi yang ditentukan. Sedangkan validasi merupakan tahapan untuk memberikan penilaian produk sesuai dan memuaskan keinginan dari pemangku kepentingan.

a. Teknik Pengujian

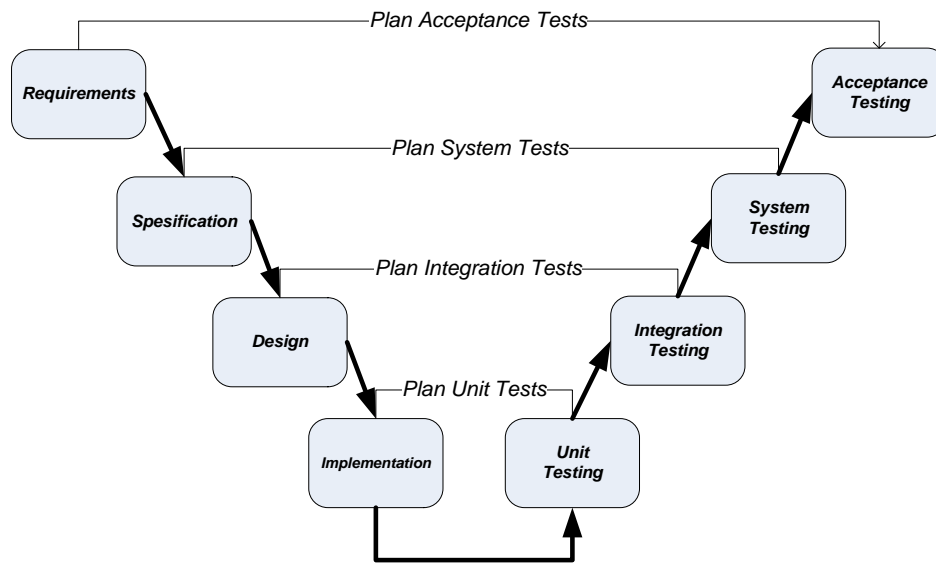
Menurut J.Watkins (2004) dalam bukunya yang berjudul *Testing IT*, menyebutkan bahwa teknik pengujian perangkat lunak dibagi menjadi tiga bagian utama seperti yang ditunjukkan pada table berikut :

Tabel 5. Teknik Pengujian Perangkat Lunak

<i>General Testing</i>	<i>Functional Testing</i>	<i>Nonfunctional Testing</i>
<ul style="list-style-type: none">- <i>Positive and negative testing</i>- <i>Whitebox and blackbox testing</i>- <i>Error guesing</i>- <i>Automated software testing</i>	<ul style="list-style-type: none">- <i>Equivalence Partitioning</i>- <i>Boundary analysis</i>- <i>Intrusive testing</i>- <i>Random testing</i>- <i>State Transition Testing</i>- <i>State testing</i>- <i>Thread testing</i>	<ul style="list-style-type: none">- <i>Configuration/Installation testing</i>- <i>Compability testing</i>- <i>Documentation and help testing</i>- <i>Fault recovery testing</i>- <i>Performance testing</i>- <i>Reliability testing</i>- <i>Security testing</i>- <i>Load testing</i>- <i>Usability testing</i>- <i>Volume testing</i>

Penggunaan dari berbagai macam pengujian ini dapat dilakukan kapan saja tergantung dengan kebutuhan, kesesuaian dan karakteristik perangkat lunak.

b. V-Model



Gambar 3. V-Model

Model ini merupakan perluasan dari model *waterfall*. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model *waterfall*. Jika dalam model *waterfall* proses dijalankan secara linear, maka dalam V-model proses dilakukan bercabang. Dalam V-model ini digambarkan hubungan antara tahap pengembangan software dengan tahap pengujiannya. Berikut penjelasan masing-masing tahap beserta tahap pengujiannya :

1) *Requirement Analysis & Acceptance Testing*

Tahap *Requirement Analysis* sama seperti yang terdapat dalam model *waterfall*. Keluaran dari tahap ini adalah dokumentasi kebutuhan pengguna. *Acceptance Testing* merupakan tahap yang akan mengkaji apakah dokumentasi yang dihasilkan tersebut dapat diterima oleh para pengguna atau tidak.

2) *Spesification & System Testing*

Dalam tahap ini analis sistem mulai merancang sistem dengan mengacu pada dokumentasi kebutuhan pengguna yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data, dan yang lain. Selain itu tahap ini juga menghasilkan contoh tampilan window dan juga dokumentasi teknik yang lain seperti *Entity Diagram* dan *Data Dictionary*.

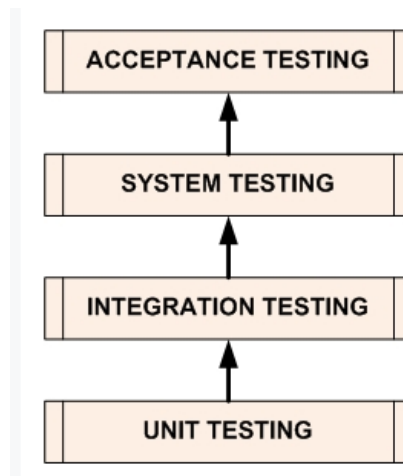
3) *Architecture Design & Integration Testing*

Sering juga disebut *High Level Design*. Dasar dari pemilihan arsitektur yang akan digunakan berdasar kepada beberapa hal seperti: pemakaian kembali tiap modul, ketergantungan tabel dalam basis data, hubungan antar interface, detail teknologi yang dipakai. Penyajian pada tahap desainpun berbagai macam, salah satunya menggunakan UML ataupun diagram.

4) *Implementation & Unit Testing*

Perancangan dipecah menjadi modul-modul yang lebih kecil. Setiap modul tersebut diberi penjelasan yang cukup untuk memudahkan *programmer* melakukan *coding*. Tahap ini menghasilkan spesifikasi program seperti: fungsi dan logika tiap modul, pesan kesalahan, proses input-output untuk tiap modul, dan lain-lain.

c. Fase Pengujian



Gambar 4. Tahap Pengujian

Terdapat empat fase pengujian yaitu :

1) *Unit Testing*

Unit testing adalah proses pengujian perangkat lunak dimana masing-masing unit/komponen diuji. Tujuannya adalah untuk memvalidasi bahwa setiap unit perangkat lunak sudah melakukan seperti apa yang telah dirancang.

Menurut Pressman, *unit testing* berfokus pada upaya verifikasi terhadap unit terkecil dari perancangan perangkat lunak. Pengujian unit berfokus pada logika pemrosesan internal dan struktur data didalam komponen.

Unit testing merupakan proses di mana pengujian dilakukan pada bagian *basic* dari kode program. Contohnya adalah memeriksa kode program pada *event*, *procedure*, dan *function*. *Unit Testing* meyakinkan bahwa masing-masing unit tersebut berjalan sebagaimana mestinya.

Pada *Unit Testing*, pengujian dilakukan dengan memeriksa bagian kode program secara terpisah dari bagian yang lain. Pengujian dapat langsung dilakukan setiap kali sebuah kode unit (*event, procedure, function*) selesai dibuat. Kode unit diperiksa dengan menjalankannya baris per baris untuk memastikan bahwa proses yang dilakukan berjalan sebagaimana yang diinginkan.

Unit testing dapat dilakukan dengan menggunakan metode *whitebox testing, functional testing, static testing, state transition testing, nonfunctional testing*.

2) *Integration Testing*

Integration Testing adalah proses pengujian perangkat lunak dimana unit individu digabungkan dan diuji sebagai sebuah kelompok. Sehingga pengujian ini mampu menampilkan kesalahan dalam interaksi antar unit

Menurut Pressman, pengujian integrasi adalah teknik untuk membangun arsitektur perangkat lunak, sementara pada saat yang sama melakukan pengujian untuk menemukan kesalahan terkait antarmuka. Tujuannya adalah untuk mengambil komponen yang diuji dan membangun struktur program yang telah ditentukan oleh perancangan.

Setelah melakukan *Unit/Component Testing*, langkah berikutnya adalah memeriksa bagaimana unit-unit tersebut bekerja sebagai suatu kombinasi, bukan lagi sebagai suatu unit yang individual. Sebagai contoh, kita memiliki sebuah proses yang dikerjakan oleh dua function, di mana satu *function* menggunakan hasil output dari *function* yang lainnya. Kedua *function* ini telah berjalan dengan baik secara individu pada *Unit Testing*.

Pada tahap *Integration Testing*, kita memeriksa hasil dari interaksi kedua *function* tersebut, apakah bekerja sesuai dengan hasil yang diharapkan. Kita juga harus memastikan bahwa seluruh kondisi yang mungkin terjadi dari hasil interaksi antar unit tersebut menghasilkan output yang diharapkan. *Integration testing* dapat dilakukan dengan pengujian *blackbox*.

3) *System Testing*

System Testing adalah proses pengujian dimana perangkat lunak yang diuji sudah lengkap dan terintegrasi. Tujuan dari pengujian ini adalah untuk mengevaluasi kesesuaian sistem dengan persyaratan yang telah ditentukan.

Menurut Pressman, pengujian sistem merupakan serangkaian pengujian yang berbeda-beda yang bertujuan untuk memverifikasi bahwa semua elemen sistem telah terintegrasi dengan baik dan menjalankan fungsi yang telah ditetapkan. *System Testing* mencakup pengujian aplikasi yang telah selesai dikembangkan. Karena itu, aplikasi harus terlihat dan berfungsi sebagaimana mestinya terhadap pengguna akhir.

System Testing dapat dilakukan menggunakan metode *Blackbox*, *nonfunctional test* (*configuration test*, *compability test*, *stress test*, *performance test* dan lain sebagainya), *thread testing*, *static testing*, *automated testing tools*.

4) *Acceptance Testing*

Acceptance Testing atau uji penerimaan adalah pengujian formal dilakukan untuk menentukan apakah sistem menerima kriteria penerimaan dan memastikan jika pengguna dapat menerima sistem. Tujuan dari pengujian ini adalah untuk mengetahui tingkat kelayakan dari perangkat lunak.

Seperti *Integration Testing*, *Acceptance Testing* juga meliputi pengujian keseluruhan aplikasi. Perbedaannya terletak pada siapa yang melakukan testing. Pada tahap ini, *end-user* yang terpilih melakukan testing terhadap fungsi-fungsi aplikasi dan melaporkan permasalahan yang ditemukan. Pengujian yang dilakukan merupakan simulasi penggunaan nyata dari aplikasi pada lingkungan yang sebenarnya. Proses ini merupakan salah satu tahap final sebelum pengguna menyetujui dan menerima penerapan sistem aplikasi yang baru. Karena itu pada tahap ini sudah tidak difokuskan untuk mengangkat permasalahan kecil seperti kesalahan pengetikan. Hal-hal minor seperti di atas sudah seharusnya ditangani selama *Unit/Component Testing* dan *Integration Testing*. *Acceptance test* dapat dilakukan menggunakan metode *Blackbox*, *usability test*, *thread testing*, *static testing*.

6. **Construct 2**

Construct 2 adalah game editor berbasis HTML 5 yang dikembangkan oleh Scirra Ltd, perusahaan yang berasal dari kota London, Inggris.



Gambar 5. Halaman Website *Construct 2*

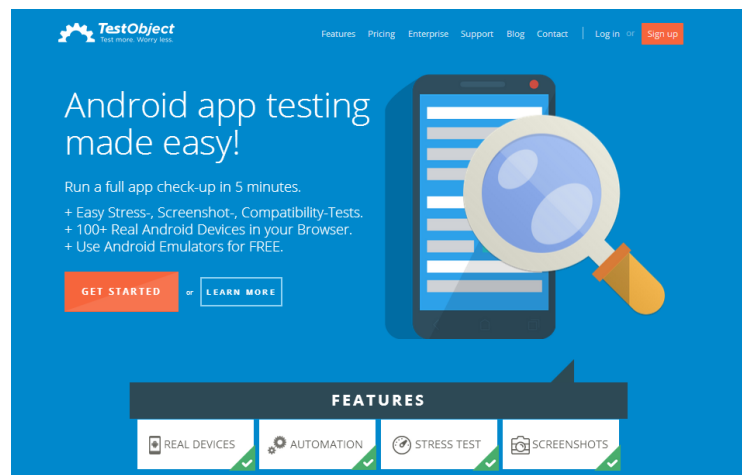
Game builder ini sebenarnya dirancang untuk game berbasis 2D. Dengan menggunakan *Construct 2*, pengembang permainan dapat mem-publishnya ke beberapa *platform* seperti HTML 5 website, Google Chrome Webstore, Facebook, Phonegap (Android), Windows Phone, Windows 8.

Pada *Construct 2* telah disediakan 70 *visual effect* yang menggunakan *engine WebGL*. Selain itu juga dilengkapi dengan 20 *built-in plugin* dan *behavior* (perilaku objek) sehingga kita bisa membuat *sprite*, objek teks, mengkoneksikan dengan facebook, menambah musik, memanipulasi penyimpanan data game dan lain sebagainya.

Pemanggilan fungsi-fungsi di *Construct 2* dilakukan dengan menggunakan pengaturan *Events* yang telah disediakan. *Events* merupakan pilihan-pilihan *action* dan kondisi yang akan menjadi nyawa dalam *game*, sehingga *game* akan berjalan sesuai dengan yang diinginkan. Karena berbasis HTML 5, maka *preview* saat *running* ketika ingin mencoba *game* dapat dilakukan pada *browser (localhost)*.

7. *Test Object*

TestObject merupakan sebuah *website* asal Jerman yang menyediakan layanan pengujian aplikasi android secara otomatis melalui jaringan *Cloud Computing* yang dimilikinya. Salah satu misi dari perusahaan ini adalah memberikan layanan bagi para pengembang perangkat lunak untuk menguji kualitas aplikasi yang dibuat agar para pengembang dapat fokus pada aplikasi yang dibuat. Oleh karena kesamaan visi ini lah yang membuat peneliti memutuskan menggunakan layanan pada *TestObject* sebagai bagian dalam pengujian aplikasi permainan Edunvi.



Gambar 6. Halaman Website *TestObject*

Cara pengujian pada *TestObject* adalah dengan mengunggah aplikasi android yang kita miliki agar kita dapat mengujinya dengan berbagai fitur yang telah disediakan. *TestObject* dapat diakses melalui www.testobject.com.

Dengan menggunakan layanan pengujian otomatis memberikan kemudahan bagi pengembang aplikasi terutama dari segi waktu pengujian

dan sumber daya. Layanan gratis yang diberikan oleh *TestObject* antara lain :

a *Real Device*

Selain menggunakan *virtual device*, *TestObject* juga memiliki *Real Device* yaitu pengujian menggunakan perangkat asli yang dimiliki pada sistem *CloudTestObject*.

b *Automation*

Merupakan fitur dimana kita bisa mengatur skenario perlakuan untuk menguji pada berbagai perangkat, sehingga nantinya skenario ini akan berjalan terus berulang-ulang pada perangkat yang berbeda.

c *Strees Test*

Menguji kehandalan perangkat lunak dalam berbagai keadaan atau kemungkinan buruk yang akan dilakukan oleh pengguna, seperti melakukan ribuan *touch* pada perangkat lunak.

d *Screenshoot*

Selama melakukan pengujian *TestObject* akan melakukan Screenshoot setiap hasil yang dilakukan, sehingga akan memudahkan bagi pengembang untuk mengamati hasil yang diperoleh.

B. Hasil Penelitian yang Relevan

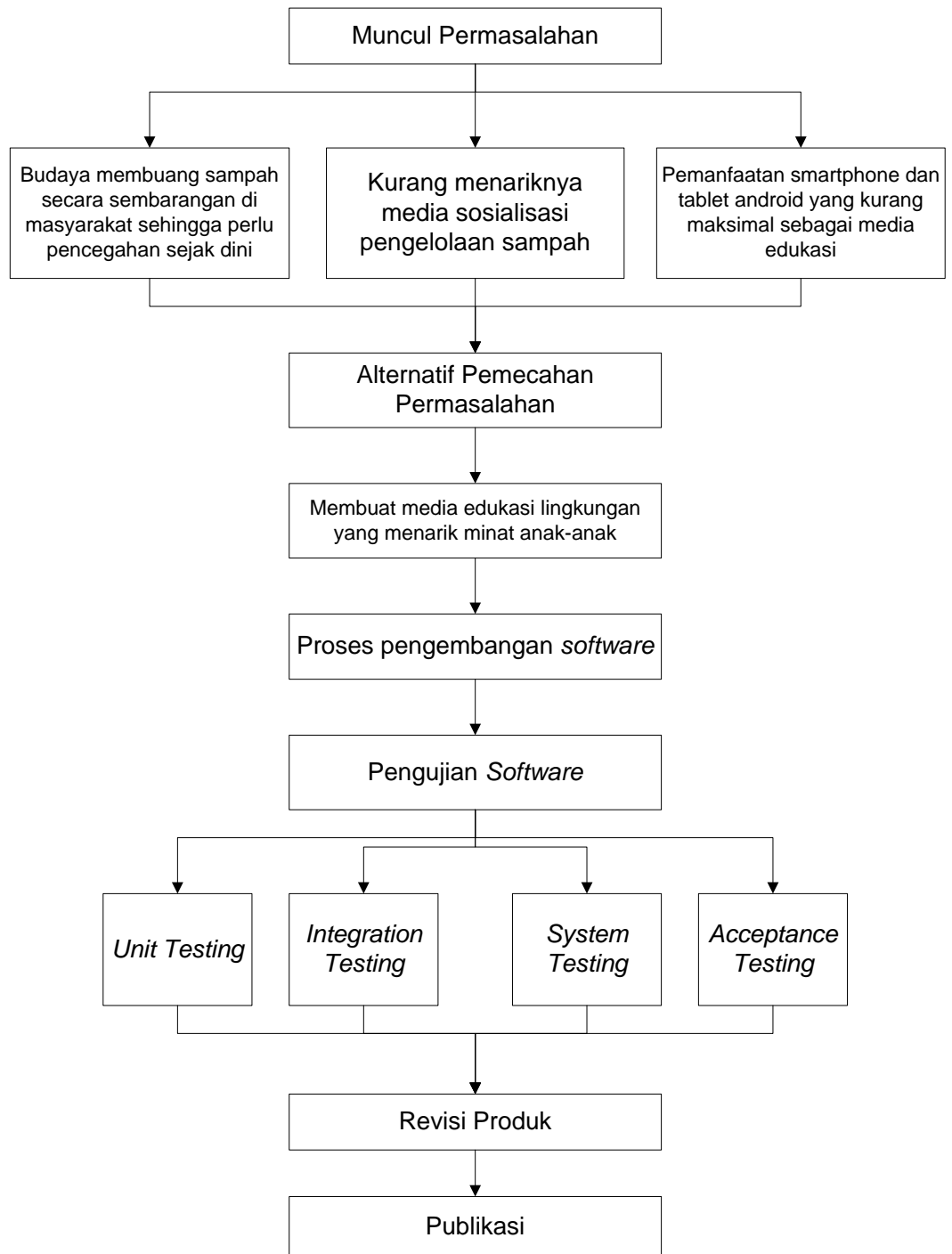
- 1 Skripsi Pengembangan Game Run In Cloud Berbasis Html5 Menggunakan Game Engine Construct 2 oleh Enlik pada tahun 2013. Penelitian ini bertujuan untuk merancang sebuah game HTML5 sebagai teknologi game masa kini yang bersifat *cross-platform*, yaitu dengan

satu basis code yang sama, game dapat dijalankan di banyak platform. Dalam penelitian ini, penulis menggunakan metode pengembangan SDLC (*System Development Life Cycle*) dengan model proses *waterfall*. Hasil yang diperoleh, Run in Cloud, game yang dikembangkan pada penelitian ini, menjadi game yang bermanfaat sebagai media hiburan bagi para pengguna baik dari kalangan *gamer* maupun *non-gamer*. *Game* ini juga memiliki kelebihan karena dapat dimainkan secara *cross-platform* yaitu pada web browser, desktop OS, dan mobile OS sehingga jangkauan pengguna menjadi lebih luas.

- 2 Skripsi Perancangan Mobile Game Sebagai Media Edukasi *Zero Waste Lifestyle* Bagi Remaja oleh Gustin A. Sutedja. Tujuan dari penelitian ini adalah untuk merancang sebuah mobile game sebagai media edukasi *Zero Waste Lifestyle* bagi remaja serta model bisnisnya dengan harapan remaja Indonesia menjadi lebih mengerti dan peduli terhadap masalah sampah di negaranya melalui game ini sehingga mereka tergerak untuk menjalani *Zero Waste Lifestyle*.

C. Kerangka Pikir

Kerangka pikir merupakan gambaran logis bagaimana variabel-variabel saling berhubungan. Kerangka pikir menjelaskan hubungan antara variable yang akan diteliti.



Gambar 7. Bagan Kerangka Pikir Penelitian

BAB III

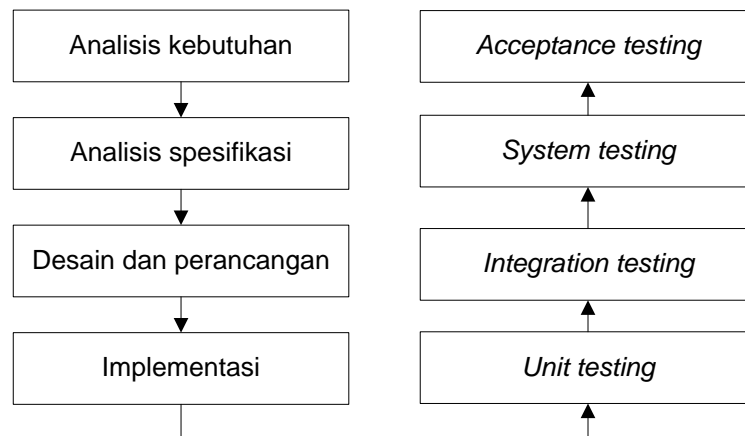
METODE PENELITIAN

A. Desain Penelitian

Metode yang digunakan dalam penelitian ini menggunakan metode Penelitian dan Pengembangan (*Reasearch and Development*). Metode Penelitian dan Pengembangan adalah metode yang digunakan untuk menghasilkan suatu produk perangkat lunak dan menguji kelayakan perangkat lunak yang dibangun. Menurut Sugiyono (2010), metode Penelitian dan Pengembangan adalah metode yang digunakan untuk menghasilkan produk tertentu dan menguji keefektifan produk tersebut.

Pada penelitian ini, peneliti membangun perangkat lunak dengan metode V-Model. Model ini merupakan perluasan dari model *waterfall*. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model *waterfall*. Jika dalam model *waterfall* proses dijalankan secara linear, maka dalam model V proses dilakukan bercabang. Dalam model V ini digambarkan hubungan antara tahap pengembangan software dengan tahap pengujiannya.

Alur penelitian yang digunakan dalam penelitian ini adalah sebagai berikut :



Gambar 8. Diagram Alur Penelitian

B. Metode Pengembangan

Pengembangan aplikasi permainan Edunvi berbasis Android pada penelitian ini menggunakan metode pengembangan perangkat lunak V-Model, dengan urutan sebagai berikut :

1. Analisis Kebutuhan

Analisis kebutuhan merupakan proses pengumpulan informasi tentang kebutuhan-kebutuhan pengguna terhadap perangkat lunak yang akan dikembangkan. Informasi ini nantinya akan digunakan sebagai acuan untuk mengetahui fitur apa saja yang akan ada pada Game Edunvi Berbasis Android.

2. Analisis Spesifikasi

Analisis spesifikasi menjabarkan apa-apa saja yang harus dimiliki oleh sistem agar dapat berjalan. Analisis spesifikasi bertujuan untuk

mengetahui sistem seperti apa yang cocok diterapkan, perangkat keras dan perangkat lunak apa saja yang dibutuhkan serta siapa saja pengguna yang akan menggunakan sistem ini.

3. Desain

a. Desain *Game*

Dalam sebuah pengembangan *game*, tahap desain *game* merupakan tahap perencanaan yang dilakukan oleh seorang pengembang untuk menentukan *genre* atau jenis *game*, cara bermain, tingkat kesukaran atau level *game*.

b. Desain Sistem dengan UML

Proses pengembangan selanjutnya adalah proses desain pengembangan perangkat lunak. Desain sistem perangkat lunak yang dibuat nantinya akan dijadikan sebagai acuan pengembang dalam penulisan kode. Desain sistem perangkat lunak ini harus sesuai karakteristik perangkat lunak yang akan dibuat serta mampu mempermudah dan memperjelas pengembang dalam proses pembuatan perangkat lunak. Penelitian ini menggunakan desain sistem model *Unified Model Language* (UML) dikarenakan model ini paling sesuai digunakan untuk mengembangkan sistem berorientasi objek.

c. Desain Antar Muka

Desain antar muka merupakan hal yang sangat penting dalam pengembangan perangkat lunak terutama dalam mengembangkan aplikasi permainan. *User interface* merupakan jembatan interaksi antara

pengguna dengan sistem, selain itu desain tokoh dan karakter dalam permainan juga harus direncanakan dengan baik dan menarik. Sehingga pengguna akan merasa nyaman dan tertarik dalam bermain *game*. Desain antar muka juga harus mampu menggambarkan nilai atau kisah yang akan disampaikan dalam bentuk permainan kepada pengguna.

4. Implementasi

Desain yang telah dirancang kemudian ditranslasikan kedalam kode melalui *event-event* untuk mengimplementasikan logika program. Proses implementasi ini dilakukan pada perangkat lunak pengembangan.

C. Metode Pengujian

Proses pengujian perangkat lunak dilakukan untuk memastikan bahwa perangkat lunak yang dikembangkan sudah berjalan dengan semestinya. Pengujian perangkat lunak dilakukan melalui empat tahap, seperti yang ditunjukkan tabel berikut :

Tabel 6. Metode Pengujian Edunvi

Teori Pressman	Teori John Watkins	Teknik Pengujian	Aspek Uji
Verifikasi	<i>Unit Testing</i>	<i>White Box</i>	<i>Functionality</i>
	<i>Integration Testing</i>	<i>Black Box</i>	
Validasi	<i>System Testing</i>	<i>Stress Testing</i>	<i>Reliability</i>
		<i>Installation Test</i>	<i>Compatibility</i>
		<i>Playability Test</i>	<i>Playability</i>

Menurut Pressman (2010) pengujian perangkat lunak adalah proses verifikasi dan validasi perangkat lunak yang diuji. Pada *Unit Testing dan Integration Testing* merupakan tahap verifikasi pengujian

perangkat lunak yang dilakukan dengan metode *whitebox* dan *blackbox*. Kemudian tahap validasi pengujian perangkat lunak yaitu pada *System Testing* dan *Acceptance Testing* dilakukan pengujian alpha untuk mengukur tingkat *playability* perangkat lunak.

D. Variabel Penelitian

Variabel penelitian yang ada dalam penelitian Edunvi game edukasi lingkungan berbasis android ini adalah :

1. *Functionality* (Fungsionalitas) merupakan kemampuan perangkat lunak untuk menyediakan fungsi sesuai kebutuhan pengguna, ketika digunakan dalam kondisi tertentu.
2. *Reliability* (Realibilitas) merupakan kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu, ketika digunakan dalam kondisi tertentu.
3. *Compability* (kompatibel) merupakan kemampuan perangkat lunak untuk ditransfer dari satu lingkungan ke lingkungan lain.
4. *Playability* (kemampuan untuk dimainkan) merupakan kemampuan perangkat lunak untuk mudah dipahami, dipelajari, digunakan, dan menarik bagi pengguna, ketika digunakan dalam kondisi tertentu.

E. Subjek Penelitian

Subyek penelitian untuk aspek *playability* adalah 20 orang pengguna aktif android. Hal ini sesuai dengan pendapat yang dikemukakan oleh Nielsen (2012) yang menyatakan jumlah sampel ideal untuk melakukan uji *usability/acceptance* sejumlah 20 orang.

Sedangkan subyek untuk menguji aspek *functionality, reliability* dan *compability* adalah Game Edunvi berbasis Android.

F. Tempat dan Waktu Pelaksanaan

Penelitian Game Edunvi dilakukan di laboratorium komputer Jurusan Pendidikan Teknik Elektronika dan dilaksanakan selama bulan Agustus.

G. Teknik Pengumpulan Data

Teknik pengumpulan data yang akan digunakan dalam penelitian ini adalah :

1. Observasi

Teknik observasi dilakukan untuk mengumpulkan data yang terkait dengan aspek *functionality, reliability* dan *compability*.

2. Kuisisioner

Teknik pengumpulan data yang dilakukan dengan cara memberikan seperangkat pertanyaan atau pernyataan tertulis kepada responden. Teknik ini dilakukan untuk mengumpulkan data yang terkait dengan aspek *playability*.

H. Instrumen Penelitian

Instrumen yang digunakan pada penelitian ini meliputi uji aspek *functionality*, *reliability*, *compatibility* dan *playability*.

1. *Functionality*

Pengujian aspek *functionality* pada aplikasi permainan Edunvi yaitu menggunakan pengujian *whitebox* dan *blackbox*. Pengujian dilakukan dengan melakukan pengamatan yang didasarkan pada *test case* yang dibuat berdasarkan *flowgraph* dan *test case* yang didasarkan pada *use case*.

2. *Reliability*

Pengujian aspek *reliability* pada aplikasi permainan Edunvi dilakukan dengan menggunakan *tools* pada website pengujian kualitas aplikasi android yaitu www.TestObject.com. Metode pengujian yang dilakukan menggunakan *StressTest* yaitu program yang mampu secara otomatis mensimulasikan ribuan tindakan pengguna dengan menggunakan *Google Monkey Exercisier* untuk menjalankan *click* atau *touch* secara acak dalam aplikasi.

3. *Compatibility*

Pengujian aspek *compatibility* pada aplikasi permainan Edunvi dilakukan dengan mengujicobakan pada emulator *Android Virtual Device* dan *Real Device* yang tersedia pada www.TestObject.com dan diatur pada sistem operasi yang berbeda dan resolusi layar yang berbeda.

a Sistem Operasi

- 1) Android Ice Cream Sandwitch (4.0.4)
- 2) Android Jelly Bean (4.3.0)
- 3) Android Kitkat (4.4.4)

b Resolusi Layar

- 1) WVGA (480x800)
- 2) FHD (1080x1920)
- 3) QHD (1600x2560)

4. *Playability*

Pengujian aspek *playability* pada aplikasi permainan Edunvi menggunakan kuisisioner yang dibuat oleh Hannu Korhonen (2006) berdasarkan kriteria dari Heuristics yang dikemukakan oleh Desurvire (2004). Desurvire menyebutkan bahwa terdapat empat faktor Playability Heuristics yaitu *Game Play, Game Story, Mechanics dan Usability*. Namun oleh Korhonen yang melakukan penelitian *Playability* Heuristic yang dikhususkan pada perangkat *mobile* menyebutkan terdapat 3 faktor yaitu *Game Play, Usability dan Mobility*. Berikut kuisisioner yang dikemukakan oleh Korhonen :

Tabel 7. Kuisioner *Playability*

No	Pertanyaan	1	2	3	4	5
Game Usability						
GU1	Penyajian audio-video mendukung permainan					
GU2	Tata letak layar efisien dan visual yang ditampilkan menyenangkan					
GU3	Perangkat dan permainan digunakan sesuai tujuannya					
GU4	Indikator-indikator terlihat jelas					
GU5	Pemain memahami istilah-istilah dalam permainan					
GU6	Navigasi disajikan secara konsisten, logis dan minimalis					
GU7	Tombol kontrol konsisten dan mengikuti aturan standar					
GU8	Kontrol permainan mudah dan fleksibel					
GU9	Permainan memberikan umpan balik pada tindakan pemain					
GU10	Pemain tidak dapat membuat kesalahan permanen (membuat aplikasi error dan tidak dapat digunakan lagi)					
GU11	Pemain tidak perlu menghafal hal-hal yang tidak perlu					
GU12	Permainan disertai petunjuk/bantuan					
Game Mobility						
MO1	Saat akan bermain game dapat dimulai dengan cepat					
MO2	Permainan mengakomodasi lingkungan					
MO3	Mampu menangani interupsi (ada panggilan masuk, sms dsb)					

Game Play					
GP1	Permainan memberikan tujuan yang jelas atau mendukung tujuan dari permainan ini dibuat				
GP2	Pemain mampu melihat kemajuannya dan membandingkan hasil sebelumnya				
GP3	Pemain mendapat hadiah yang bermakna				
GP4	Pemain yang memegang kendali				
GP5	Tantangan, strategi dan kecepatan seimbang				
GP6	Pengalaman pertama bermain menimbulkan motivasi untuk bermain lagi				
GP7	Alur cerita permainan mendukung gameplay dan bermakna				
GP8	Tidak ada tugas yang berulang dan membosankan				
GP9	Pemain dapat mengekspresikan diri				
GP10	Permainan ini mendukung gaya bermain yang berbeda				
GP11	Permainan ini tidak stagnan				
GP12	Permainan ini konsisten				
GP13	Permainan menggunakan orthogonal unit differentiation (berbeda objek berbeda tujuan)				
GP14	pemain tidak mengalami kelelahan selama bermain game				

I. Analisis Data

1. Analisis *Functionality*, *Reliability* dan *Compatibility*

Analisis aspek *functionality* dilakukan berdasarkan hasil pengamatan saat melakukan pengujian *whitebox* dan *blackbox*. Sedangkan aspek *reliability* dilakukan berdasarkan hasil laporan pengujian pada www.TestObject.com. Presentase hasil uji yang diperoleh selanjutnya dideskriptifkan dan diambil kesimpulan tentang masing-masing indikator dengan cara mengubah data kuantitatif persentase tersebut menjadi data kualitatif berpedoman pada acuan konversi nilai menurut Bloom, Madaus & Hastings (1981) menggunakan tabel berikut:

Tabel 8. Skala Penilaian Media (konversi nilai)

Presentase Pencapaian	Interpretasi
$90 \leq X$	Sangat baik
$80 \leq X < 90$	Baik
$70 \leq X < 80$	Cukup
$60 \leq X < 70$	Kurang
$X < 60$	Kurang sekali

X = skor aktual

2. Analisis Aspek *Playability*

Analisis aspek *Playability* dihitung menggunakan skala Likert. Data kuantitatif dari hasil penelitian diubah menjadi data kualitatif. Hasil dari analisis instrumen nantinya akan didapatkan skor tiap instrumen kemudian dihitung rata-rata instrumen dengan rumus :

Rumus perhitungan rata-rata instrumen

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan : \bar{x} = skor rata-rata

$\sum x$ = skor total item

n = jumlah item

Setelah mendapatkan skor rata-rata tiap instrument kemudian menghitung presentase kelayakan dengan rumus berikut ini :

Rumus perhitungan presentase

$$\text{presentase kelayakan (\%)} = \frac{\text{skor yang diobservasi}}{\text{skor yang diharapkan}} \times 100\%$$

Hasil dari presentase tersebut kemudian dicocokkan dengan predikat skala Likert.

Tabel 9. Skala Penilaian Likert

No	Presentase	Interpretasi
1	0% - 20%	Sangat Tidak Layak
2	21% - 40%	Tidak Layak
3	41% - 60%	Cukup Layak
4	61% - 80%	Layak
5	81% - 100%	Sangat Layak

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Pengembangan Perangkat Lunak

1. Analisis Kebutuhan

Setelah melakukan berbagai kajian dan pengamatan terkait fitur dan fungsi yang nantinya akan ada pada *game* Edunvi maka didapatkan hasil sebagai berikut :

1. *Game* Edunvi dimainkan oleh satu pemain atau *single player* pada perangkat android dan bersifat *offline*.
2. Edunvi merupakan aplikasi permainan, sehingga fitur utama dalam aplikasi adalah dapat memainkan sebuah permainan. Oleh karena itu membutuhkan fitur "memainkan *game*".
3. *Game* Edunvi disajikan dalam berbagai level dengan tingkat kesukaran yang berbeda. Sehingga dibutuhkan fitur untuk "Memilih Level".
4. Level yang lebih sulit dapat diakses jika pemain sudah menyelesaikan level sebelumnya, sehingga membutuhkan media penyimpanan data. Untuk melakukan penyimpanan data akan dilakukan oleh fungsi tersendiri pada "Data Skor".
5. Karena permainan ini memiliki tingkat kesukaran yang berbeda maka dibutuhkan sebuah fitur untuk melakukan reset level-level pada permainan agar pengguna dapat mengatur permainan seperti sediakala. Sehingga membutuhkan fitur "*Reset Level*".

6. Pengguna membutuhkan petunjuk dalam memainkan game sehingga dibuthkan fitur "*Help*" dalam *game*.
7. Permainan menggunakan model *drag and drop* untuk menggolongkan sampah sesuai jenisnya.

Berdasarkan pada kebutuhan fungsionalitas diatas, maka dilakukan pengamatan lebih lanjut untuk mencari data mengenai konsep, teori dan perangkat untuk mendukung pengembangan perangkat lunak tersebut. Hasil yang didapatkan antara lain :

1. *Game engine* yang digunakan adalah *Construct 2* yang juga termasuk kedalam kategori "*point and click*" engine. Karena memiliki tingkat kemudahan yang tinggi serta sudah mencakup semua *library* yang digunakan termasuk fitur untuk melakukan *drag and drop*.
2. Menggunakan CocoonJS sebagai *native compiler* yang mendukung berbagai platform salah satunya sistem operasi Android.
3. Menggunakan layanan *WebStorage* sebagai media penyimpanan data yang sudah tersedia pada Construct 2.

2. Analisis Spesifikasi

a. Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan peneliti untuk mengembangkan aplikasi ini adalah sebagai berikut :

Tabel 10. Spesifikasi *Hardware* untuk Pengembangan

No	Perangkat	Spesifikasi
1	Processor	Intel Core I-3
2	RAM	2048MB
3	Hardisk	320 GB
4	VGA	Nvidia GF310M

Sedangkan untuk menentukan spesifikasi minimal dan rekomendasi untuk menjalankan aplikasi *game* yang dibuat peneliti melakukan kajian berdasarkan data dari phonearena yang bersumber dari Google terkait tingkat distribusi penggunaan sistem operasi android. Penelitian ini dilakukan pada bulan Mei 2014.

Version	Codename	API	Distribution
2.2	Froyo	8	1.0%
2.3.3 - 2.3.7	Gingerbread	10	16.2%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	13.4%
4.1.x	Jelly Bean	16	33.5%
4.2.x		17	18.8%
4.3		18	8.5%
4.4	KitKat	19	8.5%

*Data collected during a 7-day period ending on May 1, 2014.
Any versions with less than 0.1% distribution are not shown.*

Gambar 9. Distribusi Sistem Operasi Android

Berdasarkan data yang dikemukakan oleh Alan F (2014) melalui phonearena terkait tingkat distribusi sistem operasi Android, peneliti menentukan spesifikasi minimal dan rekomendasi untuk menjalankan Game Edunvi nantinya yaitu :

Tabel 11. Spesifikasi Edunvi

Perangkat	Spesifikasi
Spesifikasi Minimal	
Sistem Operasi	Android ICS
<i>Processor</i>	Dual Core
RAM	512 MB
<i>Memory</i>	25 MB
Spesifikasi Rekomendasi	
Sistem Operasi	Jelly Bean
<i>Processor</i>	QuadCore
RAM	1024 MB
<i>Memory</i>	30 MB

Penentuan spesifikasi tersebut, dilihat dari banyaknya perangkat yang saat ini menggunakan spesifikasi tersebut. Oleh karena itu diharapkan Game Edunvi mampu dijalankan dengan baik pada rentang spesifikasi tersebut.

b. Analisis Kebutuhan Perangkat Lunak

Pada pembuatan aplikasi ini membutuhkan beberapa perangkat lunak sebagai media pengembangan. Perangkat lunak yang digunakan antara lain :

1. Construct 2
2. Adobe Illustrator
3. Adobe Photoshop
4. Corel Draw
5. Audacity (*Audio Editor*)

3. Desain

a. Desain Game

1) Genre Permainan

Game Edunvi adalah jenis permainan edukasi dengan grafis 2D dengan genre *arcade*, sehingga membutuhkan ketangkasan pengguna dalam memainkan permainan.

2) Menentukan *Gameplay*

Gameplay atau cara bermain dari permainan ini adalah pemain diharuskan menggolongkan sampah-sampah yang berjatuhan sesuai jenisnya ke dalam tempat sampah yang disediakan. Penggolongan sampah dilakukan dengan metode *drag and drop*.

Setiap tindakan yang dilakukan pengguna akan mempengaruhi tampilan lingkungan sekitar atau *background* yang ditampilkan.

3) Menentukan Level

Game Edunvi dibuat dengan tiga level yang berbeda dengan karakteristik dan tingkat kesulitan yang berbeda pula. Perbedaan tiap level dalam permainan ini adalah sebagai berikut :

Tabel 12. Deskripsi Level pada Edunvi

Tingkatan Level	Deskripsi
Level 1	Terdapat 1 Baris sampah jatuh, dan dipilah-pilah kedalam sampah organik dan anorganik dengan kecepatan gerak 5 px per detik
Level 2	Terdapat 2 Baris sampah jatuh, dan dipilah-pilah kedalam sampah organik, anorganik dan plastik dengan kecepatan gerak 5 px per detik.
Level 3	Terdapat 2 Baris sampah jatuh, dan dipilah-pilah kedalam sampah organik, anorganik, logam dan plastik dengan kecepatan gerak 10 px per detik.

b. Desain Sistem dengan UML

Proses pengembangan selanjutnya adalah proses desain pengembangan perangkat lunak. Desain sistem perangkat lunak yang dibuat nantinya akan dijadikan sebagai acuan pengembang dalam penulisan kode. Desain sistem perangkat lunak ini harus sesuai karakteristik perangkat lunak yang akan dibuat serta mampu mempermudah dan memperjelas pengembang dalam proses pembuatan perangkat lunak. Penelitian ini menggunakan desain sistem model *Unified Model Language* (UML) dikarenakan model ini paling sesuai digunakan untuk mengembangkan sistem berorientasi objek.

1) Use Case Diagram

Use case menggambarkan bagaimana seorang pengguna berinteraksi dengan sistem dengan cara menentukan langkah-langkah yang diperlukan untuk mencapai suatu tujuan tertentu. Variasi-variasi dalam urutan langkah yang terjadi antara aktor dan sistem merupakan sebuah skenario. Diagram *use case* memberikan suatu gambaran besar tentang fungsionalitas yang

diharapkan dari sebuah sistem dilihat dari sudut pandang pengguna.

Diagram use case pada pengembangan Game Edunvi terdiri dari delapan *use case*, yaitu *use case* memulai aplikasi, *use case* membuka play, *use case* membuka help, *use case* mereset level, *use case* memilih level, *use case* memainkan game, *use case* data skor dan *use case* menutup aplikasi.

Relasi antar *use case* yang berlabel *extend* terdapat pada *use case* membuka play dengan *use case* memilih level, *use case* membuka level dengan *use case* mereset level dan *use case* memilih level dengan *use case* memainkan game. Relasi ini berarti menandakan bahwa *use case* yang ditunjukkan oleh tanda panah belum tentu mempengaruhi *use case* yang dituju. Sedangkan relasi antar *use case* yang berlabel *include* terdapat pada *use case* mereset level dengan *use case* data skor, *use case* memilih level dengan *use case* data skor dan *use case* memainkan game dengan *use case* data skor. Relasi ini menandakan bahwa *use case* yang ditunjukkan panah akan mempengaruhi *use case* asal.

Definisi *use case* pada game Edunvi adalah sebagai berikut:

a) Definisi Aktor

Tabel 13. Definisi Aktor

No	Aktor	Deskripsi
1	Pengguna	Pengguna merupakan aktor dari perangkat lunak. Pengguna dapat membuka aplikasi, membuka <i>Help</i> , memilih level, melakukan <i>reset level</i> dan keluar dari aplikasi

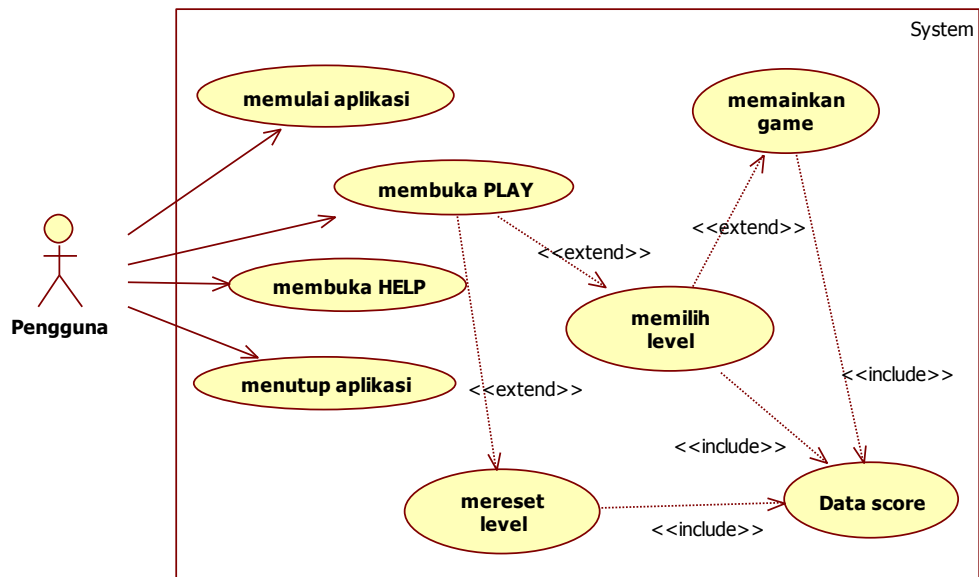
b) Definisi *Use Case*

Tabel 14. Definisi *Use Case*

No	<i>Use case</i>	Deskripsi	Relasi
1	Membuka Aplikasi	Proses Awal ketika pengguna membuka aplikasi akan ditampilkan <i>Splash Screen</i> dan kemudian akan ditampilkan halaman Menu	Aktor berinteraksi langsung dengan <i>use case</i> membuka aplikasi sehingga terdapat relasi <i>assosiation</i> diantara keduanya
2	Membuka <i>Play</i>	Merupakan menu untuk memulai permainan, ketika pengguna memilih <i>Play</i> maka akan menampilkan Intro terlebih dahulu	<i>Use case</i> membuka <i>play</i> menampilkan opsi pilihan level dan fitur untuk melakukan <i>reset level</i> sehingga memiliki relasi <i>extend</i> dengan <i>use case</i> memilih level dan use case mereset level. Use case memilih level menampilkan opsi berbagai pilihan level yang tersedia untuk dipilih sehingga memiliki relasi <i>extend</i> dengan <i>use case</i> memainkan <i>game</i>
3	Membuka <i>Help</i>	Merupakan menu yang menampilkan petunjuk dan informasi ke pengguna	Aktor berinteraksi langsung dengan <i>use case</i> membuka <i>help</i> sehingga terdapat relasi <i>assosiation</i> diantara keduanya
4	Memilih Level	Melakukan inisialisasi pada <i>database</i> untuk melakukan pengecekan skor, untuk menentukan level-level yang sudah terbuka. Kemudian pengguna bisa memilih level yang sudah terbuka	Ketika <i>use case</i> memilih level diakses perangkat lunak harus melakukan pengecekan data terlebih dahulu pada <i>use case</i> data <i>score</i> sehingga use case memilih level memiliki relasi <i>include</i> dengan <i>use case data score</i>

5	Melakukan <i>Reset Level</i>	Pengguna dapat melakukan reset level, ketika fitur ini dipanggil maka akan merubah nilai skor pada <i>database</i> menjadi nol	Ketika use case reset level diakses perangkat lunak harus melakukan pengecekan data terlebih dahulu pada <i>use case data score</i> sehingga <i>use case</i> reset level memiliki relasi <i>include</i> dengan <i>use case data score</i>
6	Memainkan <i>Game</i>	Proses ketika pengguna memainkan game Edunvi berdasarkan level yang dipilih, diakhir permainan sistem menyimpan skor pada <i>database</i>	Ketika <i>use case</i> memainkan <i>game</i> diakses perangkat lunak harus melakukan pengecekan data terlebih dahulu pada <i>use case data score</i> sehingga <i>use case</i> memainkan game memiliki relasi <i>include</i> dengan <i>use case data score</i>
7	<i>Data Score</i>	Proses menyimpan dan memuat data skor	<i>Use case data score</i> selalu terpicu oleh <i>use case</i> lain
8	Keluar Aplikasi	Proses keluar dan menutup aplikasi	aktor berinteraksi langsung dengan <i>use case</i> keluar aplikasi sehingga terdapat relasi <i>association</i> diantara keduanya

Gambar *use case* pada game Edunvi berbasis Android pada penelitian ini sebagai berikut :



Gambar 10. *Use Case* Edunvi

Penjabaran *use case diagram* diatas akan didefinisikan lebih lanjut pada tabel skenario *use case* berikut :

Nama use-case : Membuka Aplikasi
Skenario :

Tabel 15. Skenario Membuka Aplikasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pengguna Menjalankan Aplikasi	
	a. Menjalankan <i>splash screen</i> b. Menuju ke menu aplikasi

Nama use-case : Membuka *Help*
 Skenario :

Tabel 16. Skenario Membuka *Help*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pengguna Membuka <i>Help</i>	
	a. Menampilkan informasi dan petunjuk permainan

Nama use-case : Membuka *Play*
 Skenario :

Tabel 17. Skenario Membuka *Play*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Pengguna menekan <i>Play</i>	
	a. Menampilkan intro/cerita tentang permainan hingga selesai b. Menuju halaman memilih level
Skenario Alternatif	
1. Pengguna menekan skip pada saat intro berlangsung	a. Menuju halaman memilih level

Nama use-case : Memilih Level
 Skenario

Tabel 18. Skenario Memilih Level

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memilih level	
	a. Melakukan pengecekan skor pada database b. Menampilkan pilihan level yang sudah terbuka
2. Memilih level yang sudah terbuka	a. Menampilkan petunjuk permainan b. Menuju ke permainan dengan level yang dipilih

Nama use-case : Melakukan Reset Level
 Skenario :

Tabel 19. Skenario *Reset Level*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Menekan Reset Level	
	a. Menampilkan pesan peringatan b. Jika ya, merubah nilai skor pada database menjadi nol
Skenario Alternatif	
1. Menekan <i>Reset Level</i>	a. Menampilkan pesan peringatan b. Jika tidak, tidak melakukan perubahan skor pada <i>database</i>

Nama use-case : Memainkan Game
 Skenario :

Tabel 20. Skenario Memainkan *Game*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Memainkan <i>Game</i>	a. Menampilkan permainan sesuai level yang dipilih b. Jika menang melakukan <i>update</i> skor pada database c. Menampilkan pesan menang
2. Menekan <i>Exit</i>	a. <i>Update</i> skor pada database b. Keluar dari permainan
Skenario Alternatif	
1. Menekan <i>Exit</i> saat permainan berjalan	a. <i>Update</i> skor pada database b. Keluar dai permainan c. Menuju halaman menu

Nama use-case : Data *Score*
Skenario :

Tabel 21. Skenario Data *Score*

Aksi Aktor	Reaksi Sistem
Skenario Normal	
1. Data <i>score</i>	
	a. Mengakses database

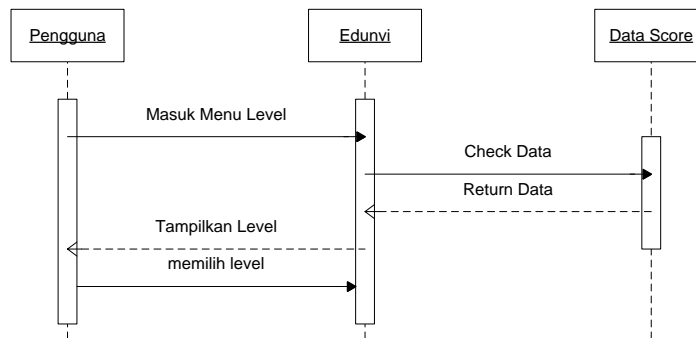
Nama use-case : Keluar Aplikasi
Skenario :

Tabel 22. Skenario Keluar Aplikasi

Aksi Aktor	Reaksi Sistem
Skenario Normal	
2. Keluar Aplikasi	
	b. Proses menutup aplikasi

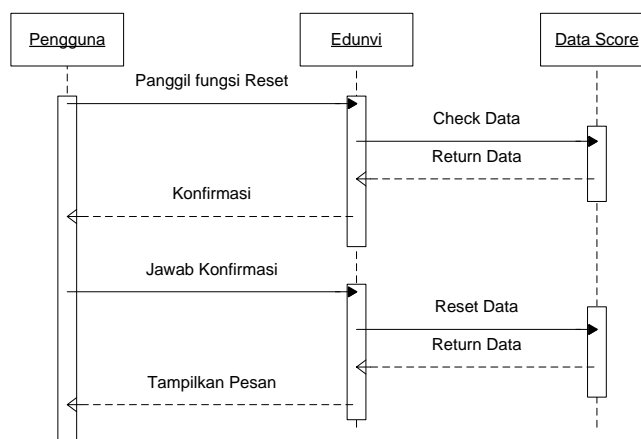
2) *Sequence Diagram*

Dalam *sequence diagram*, aktor mewakili pengguna, kotak berlabel mewakili sistem yang terotomasi, garis putus-putus vertikal sebagai perpanjangan objek, dan anak panah mewakili pesan yang dikirim ataupun diterima. Fungsi dari *diagram sequence* sendiri untuk memperjelas alur, keterkaitan dan respon dari tiap bagian *use case*. Berikut ini *sequence diagram* pada perangkat lunak yang dikembangkan :



Gambar 11. *Diagram Sequence* memilih level

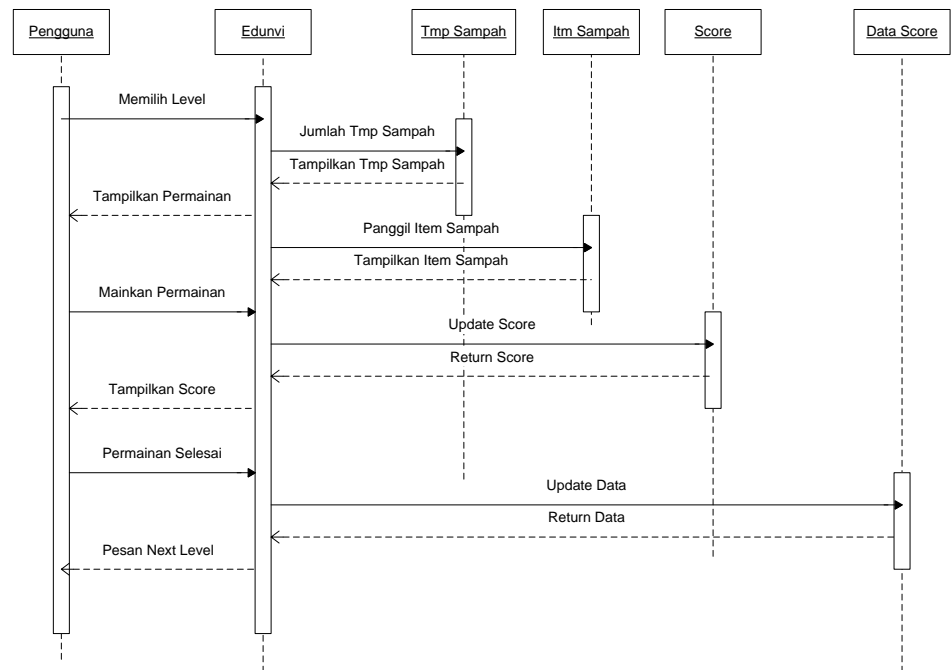
Ketika pengguna memilih menu pilih level, sistem pada Edunvi akan mengakses data skor untuk melakukan pengecekan data. Jika data yang diakses ada, maka *database* akan memberikan data tersebut pada sistem Edunvi untuk menentukan level yang dapat diakses oleh pengguna.



Gambar 12. *Diagram Sequence* Reset Skor

Ketika pengguna memilih menu reset skor, sistem pada Edunvi akan melakukan pengecekan data pada *database*.

Kemudian data yang diakses akan diberikan pada sistem Edunvi. Sistem pada Edunvi akan menampilkan konfirmasi penghapusan data pada pengguna. Respon dari pengguna terhadap pesan akan digunakan sebagai penentu data akan dihapus atau tidak.



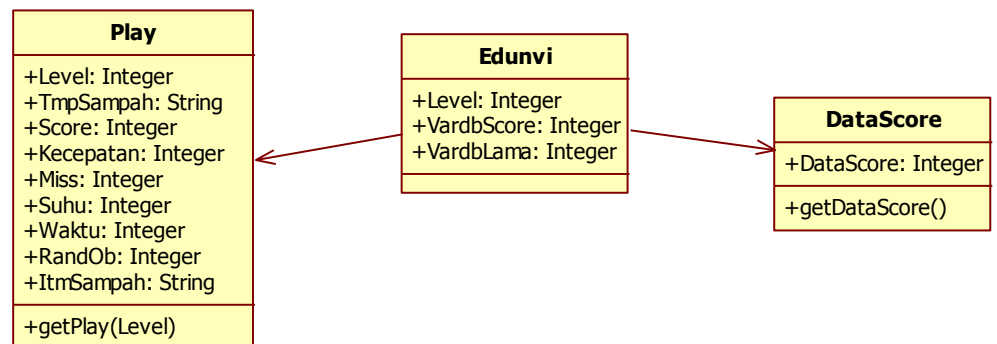
Gambar 13. *Diagram Sequence Memainkan Game*

Setelah pengguna memilih level, sistem Edunvi akan menampilkan sebuah permainan. Sistem pada Edunvi akan menampilkan jumlah tempat sampah sesuai dengan level yang diakses. Kemudian sistem akan menampilkan *item-item* sampah yang berjatuhan untuk disortir oleh pengguna pada tempat sampah. Setiap *item* sampah yang benar maka akan menambah nilai skor. Diakhir permainan sistem Edunvi akan melakukan

update skor pada database guna menentukan level yang dapat diakses selanjutnya.

3) *Class Diagram*

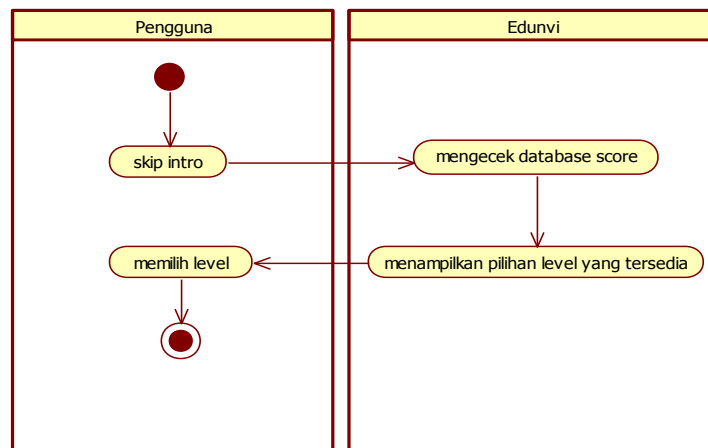
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat lengkap dengan atribut kelas dan metode-metode yang ada pada kelas tersebut.



Gambar 14. *Class Diagram* Edunvi

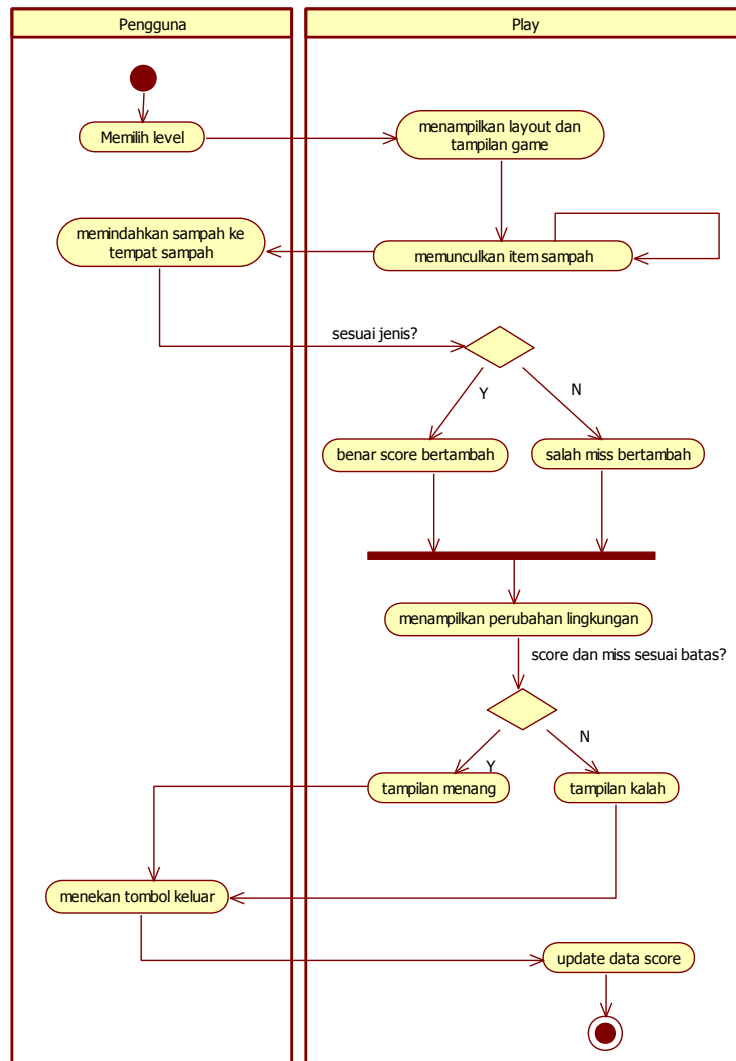
4) *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, percabangan yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut ini adalah *activity diagram* permainan Edunvi :



Gambar 15. *Activity Diagram* Memilih Level

Saat pengguna menekan tombol *skip*, sistem pada *edunvi* akan melakukan pengecekan data skor pada *database* kemudian data tersebut digunakan untuk menentukan level pilihan yang tersedia. Selanjutnya pengguna dapat memilih level yang tersedia. Proses selanjutnya adalah ketika pengguna memilih level akan dijabarkan melalui diagram berikut :



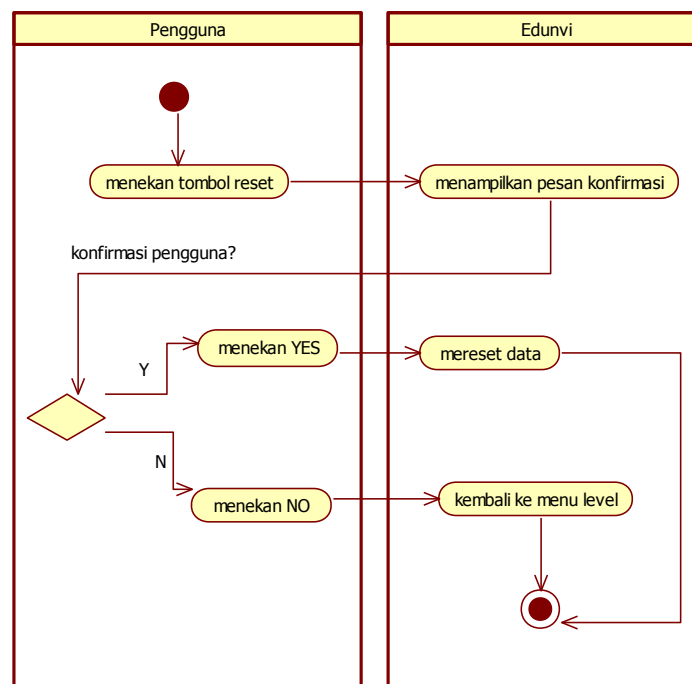
Gambar 16. *Activity Diagram* Memainkan Permainan

Setelah pengguna memilih level, sistem Edunvi akan menampilkan permainan yang sesungguhnya, yaitu dengan menampilkan *asset-asset* yang digunakan dalam permainan. Salah satu *asset* penting dalam permainan adalah item sampah yang berjatuhan secara acak dan terus menerus. Pengguna akan berinteraksi dengan item sampah ini, yaitu dengan melakukan *drag and drop* untuk diletakkan pada tempat sampah.

Setiap interaksi pengguna dengan *item* sampah ini akan dilakukan pengecekan apakah tindakan yang dilakukan pengguna ini termasuk kategori benar atau salah. Setiap tindakan pengguna, sistem akan menampilkan efek-efek sebagai umpan balik dari pengguna. Salah satunya adalah melalui penambahan skor dan perubahan lingkungan.

Ketika pengguna memenuhi kondisi menang maka akan ditampilkan halaman menang. Jika pengguna memenuhi kondisi kalah maka akan ditampilkan halaman *game over*. Diakhir permainan sistem akan melakukan *update* data.

Salah satu fitur yang terdapat pada sistem Edunvi adalah melakukan reset skor, seperti yang ditunjukkan diagram berikut :



Gambar 17. *Activity Diagram* Reset Level

Saat pengguna menekan tombol *reset*, sistem Edunvi akan menampilkan halaman peringatan dan konfirmasi tentang penghapusan data. Respon dari pengguna akan dijadikan sebagai acuan keputusan penghapusan data skor pada sistem.

c. Desain Antar Muka

Desain antar muka merupakan hal yang sangat penting dalam pengembangan perangkat lunak terutama dalam mengembangkan aplikasi permainan. *User interface* merupakan jembatan interaksi antara pengguna dengan sistem, selain itu desain tokoh dan karakter dalam permainan juga harus direncanakan dengan baik dan menarik. Sehingga pengguna akan merasa nyaman dan tertarik dalam bermain *game*.

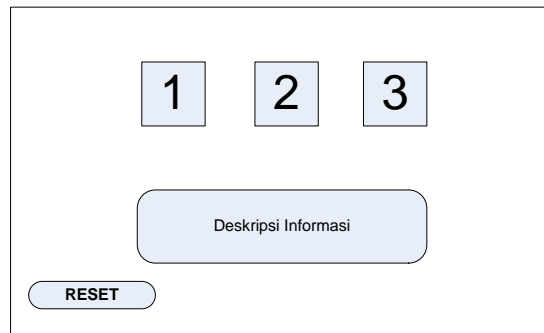
Desain antar muka juga harus mampu menggambarkan nilai atau kisah yang akan disampaikan dalam bentuk permainan kepada pengguna. Desain antar muka pada game Edunvi antara lain seperti berikut :

1) Halaman Utama



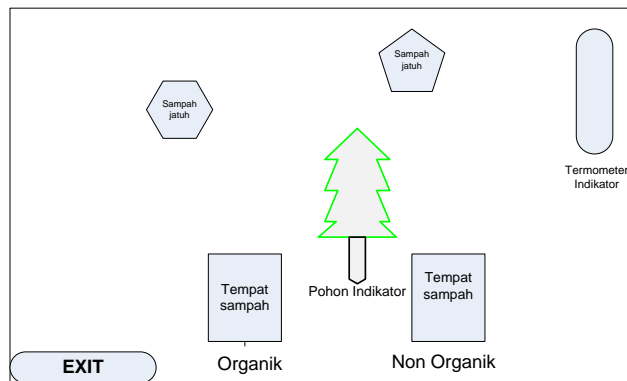
Gambar 18. Rancangan Tampilan Menu

2) Halaman Level



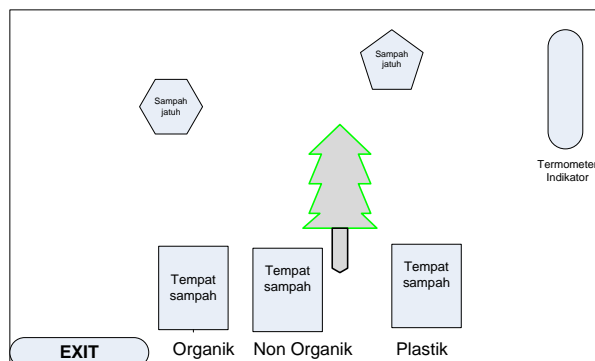
Gambar 19. Rancangan Tampilan *Level*

3) Permainan Level 1



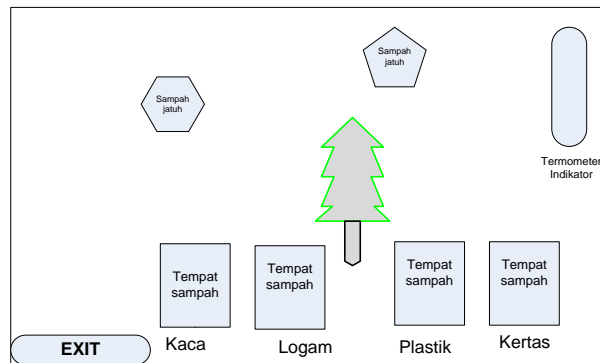
Gambar 20. Rancangan Tampilan *Level/ 1*

4) Permainan Level 2



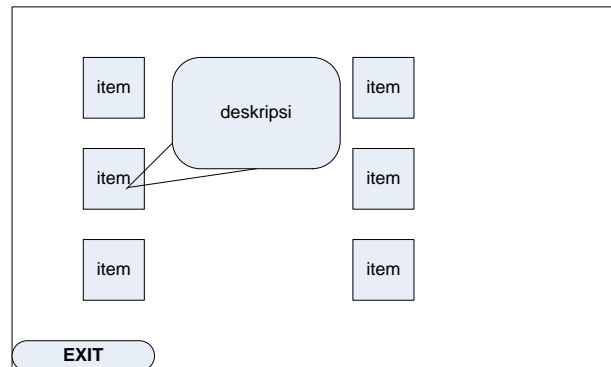
Gambar 21. Rancangan Tampilan *Level/ 2*

5) Permainan Level 3



Gambar 22. Rancangan Tampilan *Level/3*

6) Halaman Help

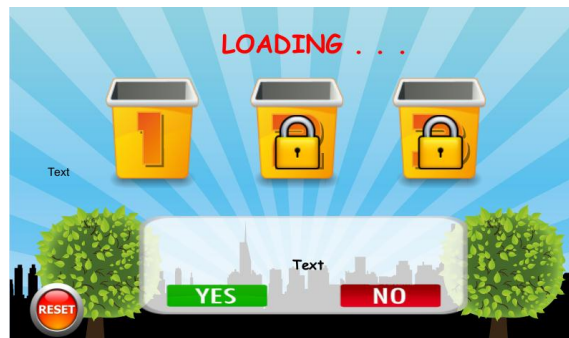


Gambar 23. Rancangan Tampilan *Help*

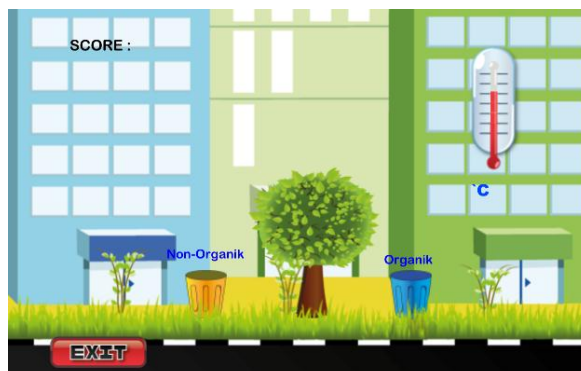
Hasil rancangan desain antarmuka yang telah dibuat tersebut kemudian dijadikan acuan dalam melakukan olah gambar. Hasil desain yang telah dibuat adalah sebagai berikut :



Gambar 24. Tampilan Menu Edunvi



Gambar 25. Tampilan Pilih Level Edunvi



Gambar 26. Tampilan *Level* 1 Edunvi



Gambar 27. Tampilan *Level/2* EduNvi



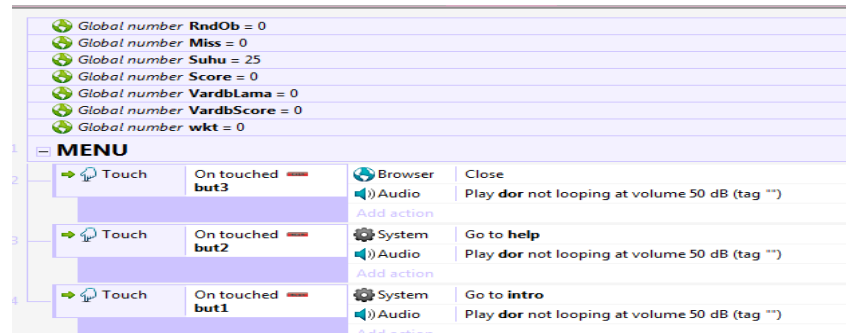
Gambar 28. Tampilan *Level/3* EduNvi



Gambar 29. Tampilan *Help* EduNvi

4. Implementasi

Desain yang telah dirancang kemudian ditranslasikan kedalam kode melalui event-event untuk mengimplementasikan logika program. Implementasi logika dan event yang dilakukan adalah sebagai berikut :



Gambar 30. Implementasi *Event*

Implementasi *event* yang lebih lengkap dapat dilihat pada halaman lampiran.

B. Pengujian Perangkat Lunak

1. *Unit Testing*

Unit Testing merupakan tahap pertama pengujian perangkat lunak, dimana pengujian dilakukan menggunakan metode *whitebox* untuk menguji aspek *fungisionaliti* perangkat lunak per unit.

a. Pengujian Whitebox

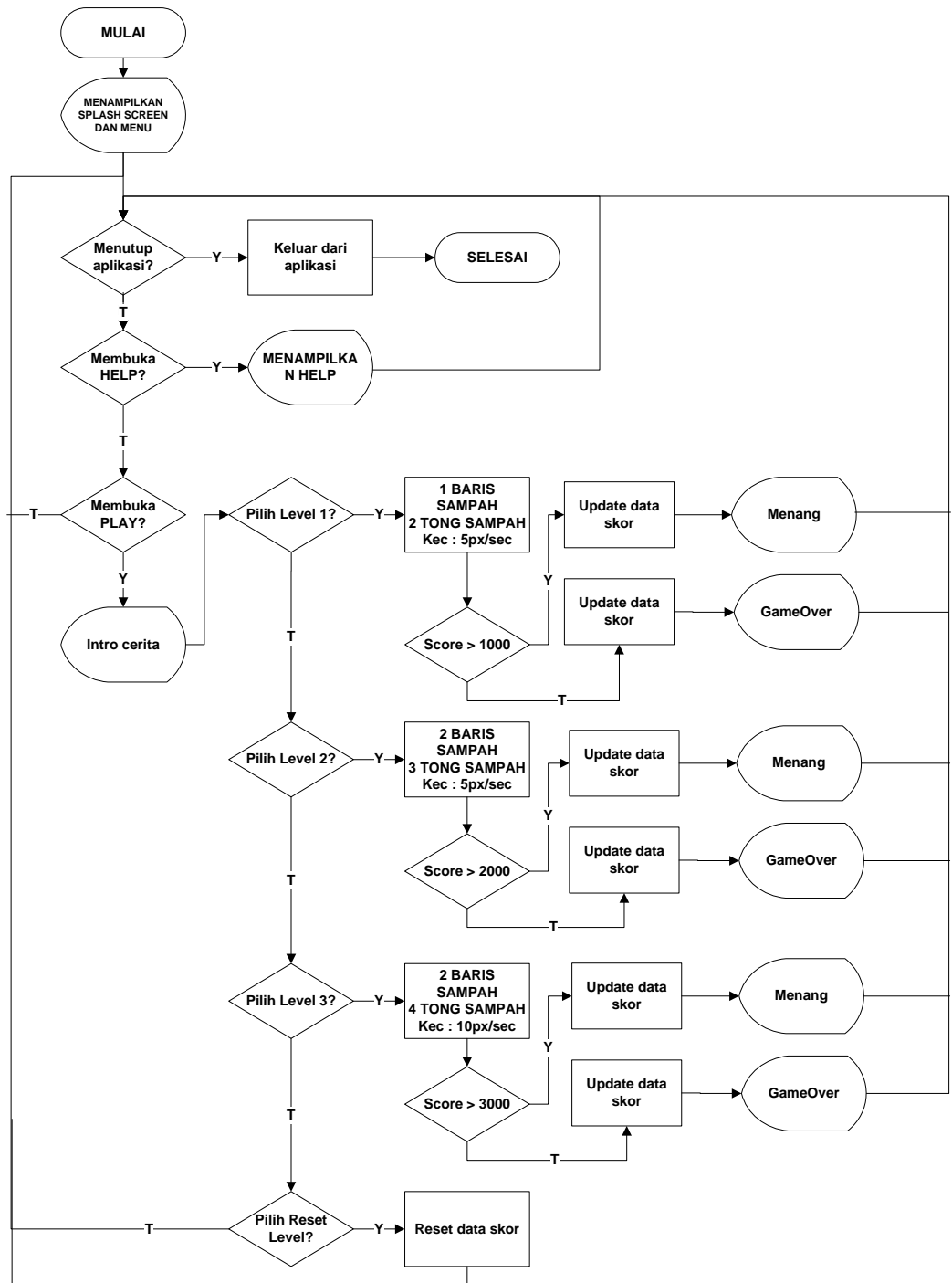
White Box Testing adalah metode desain *test case* menggunakan struktur kontrol desain prosedural untuk mendapatkan *test case*. Salah satu metode pengujian *whitebox* adalah melalui *Basis Path* Pengujian yang pertama kali diajukan oleh Tom McCabe (1976). Metode ini memungkinkan perancangan *test case* untuk menurunkan ukuran

kompleksitas logis dari suatu rancangan prosedural dan menggunakannya sebagai pedoman untuk menentukan rangkaian dasar jalur eksekusi.

Berikut hasil dari pengujian *Whitebox* :

1) Menentukan *Flowgraph*

Flowgraph atau notasi grafik alir merupakan notasi sederhana yang mempresentasikan aliran kontrol. Penyusunan *Flowgraph* didasarkan pada *Flowchart* Program.

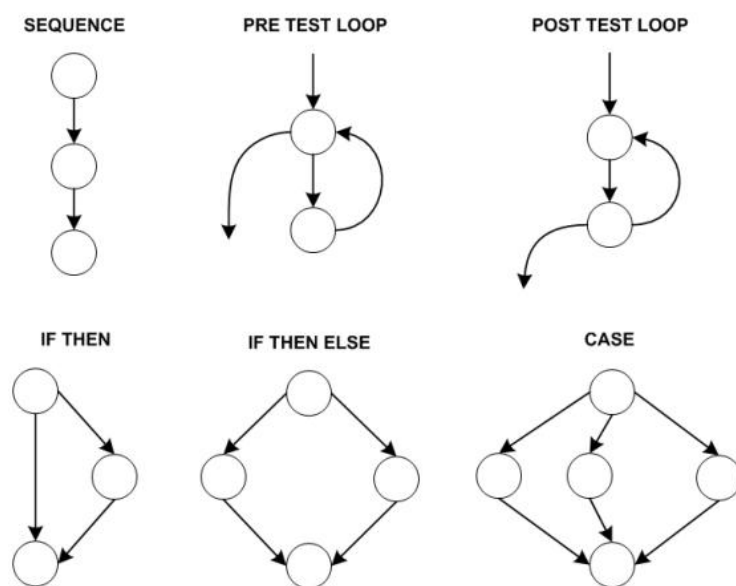


Gambar 31. *Flowchart* Edunvi

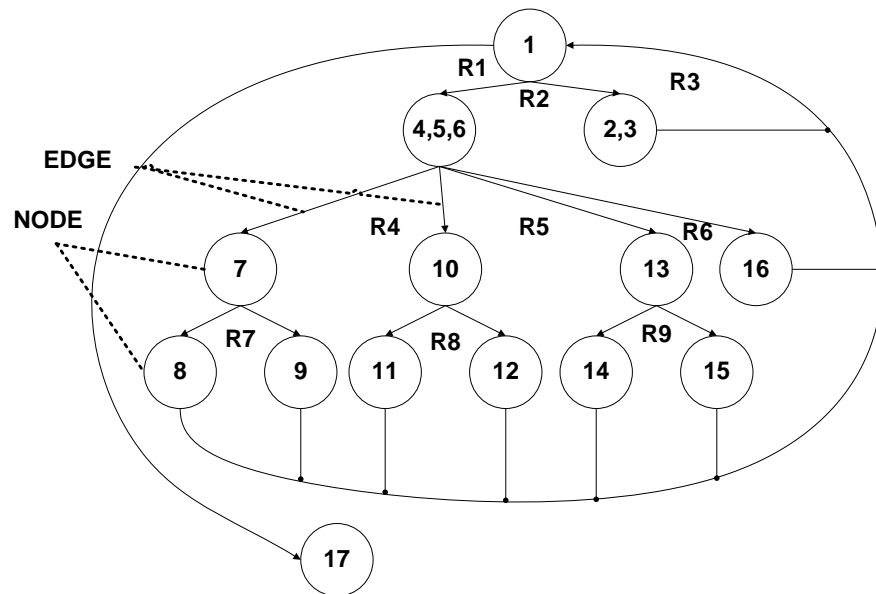
Flowgraph merupakan gambar berarah dimana node atau titik dapat merupakan keseluruhan atau potongan suatu pernyataan program. *Edge* merupakan representasi dari aliran kontrol. Region merupakan daerah yang dibatasi oleh node dan *edge*.

Menurut Pressman (2010: 589) menyebutkan bahwa urutan kotak-kotak proses dan berlian-berlian keputusan bisa dipetakan dalam satu node. Sedangkan sebuah *edge* harus berhenti di sebuah node.

Berikut ini adalah notasi *flowgraph* yang merepresentasikan kontrol logika yang digunakan menggambarkan struktur program:



Gambar 32. Notasi *Flowgraph*



Gambar 33. *Flowgraph* Edunvi

Keterangan Node *Flowgraph* Edunvi :

- 1) Mulai
- 2) Menu help
- 3) Tampilan help
- 4) Menu Play
- 5) Menampilkan intro cerita
- 6) Menu memilih level
- 7) Kondisi IF pada permainan level satu. Jika menang menuju node 8, jika kalah menuju node 9.
- 10) Kondisi "IF" pada permainan level dua. Jika menang menuju node 11, jika kalah menuju node 12.
- 13) Kondisi "IF" pada permainan level tiga. Jika menang menuju node 14, jika kalah menuju node 15.
- 16) Reset level
- 17) Keluar

2) Menghitung *Cyclomatic Complexity*

Cyclomatic Complexity atau kompleksitas sistematis adalah metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Nilai yang dihitung kompleksitas sistematis mendefinisikan jumlah *Independent Path* dalam *Basis Path Testing*, serta menyediakan batas atas untuk jumlah pengujian yang harus dilakukan guna memastikan bahwa semua pernyataan telah dieksekusi minimal sekali.

Berdasarkan teori Graph, kompleksitas sistematis dapat dihitung melalui tiga cara yaitu :

1. Jumlah Region atau daerah-daerah pada *Flowgraph*.
2. Menggunakan persamaan :

$$V(G) = E - N + 2$$

Dimana E adalah jumlah edge dan N adalah jumlah node

3. Menggunakan persamaan :

$$V(G) = P + 1$$

Dimana P adalah jumlah node yang berperingkat atau percabangan.

Dengan tiga cara penghitungan kompleksitas sistematis diatas, maka menghasilkan :

1. Berdasarkan *Flowgraph* memiliki sembilan (R) atau region
2. Diketahui : Edge = 21; Node = 14

$$V(G) = E - N + 2$$

$$V(G) = 21 - 14 + 2 = 9$$

3. Diketahui : $P = 8$

$$V(G) = P + 1$$

$$V(G) = 8 + 1 = 9$$

Menurut McCabe (1976), nilai kompleksitas sistematis yang tinggi maka akan menunjukkan prosedur yang sulit untuk dipahami, diuji, dan dipelihara. Hubungan antara kompleksitas sistematis dan resiko dalam prosedur ditunjukkan pada tabel berikut :

Tabel 23. Hubungan Kompleksitas dan Resiko

CC	Type of Procedure	Risk
<i>1-4</i>	<i>A simple procedure</i>	<i>Low</i>
<i>5-10</i>	<i>A well structured and stable procedure</i>	<i>Low</i>
<i>11-20</i>	<i>A more complex procedure</i>	<i>Moderate</i>
<i>21-50</i>	<i>A complex procedure, alarming</i>	<i>High</i>
<i>>50</i>	<i>An error-prone, extremely troublesome, untestable procedure</i>	<i>Very High</i>

Berdasarkan penghitungan diatas maka Game Edunvi menghasilkan nilai kompleksitas sistematis sebesar 9. Hal ini menunjukkan bahwa harus ada 9 kali *test case* agar semua fungsi dapat dieksekusi minimal satu kali. Selain itu berdasarkan tabel diatas menunjukkan bahwa game Edunvi terstruktur dengan baik dan memiliki tingkat resiko kegagalan rendah

3) Menentukan *Independent Path*

Independent Path adalah jalur yang dilalui program yang merepresentasikan sebuah kondisi baru. Bila dinyatakan dalam *Flowgraph* maka *Independent Path* harus bergerak setidaknya satu edge yang belum dilalui. *Independent Path* ini nantinya akan digunakan untuk melakukan uji test case. Berdasarkan *Flowgraph* diatas maka menghasilkan :

Path 1 : 1-17

Path 2 : 1-2-3-1-17

Path 3 : 1-4-5-6-7-8-1-17

Path 4 : 1-4-5-6-7-9-1-17

Path 5 : 1-4-5-6-10-11-1-17

Path 6 : 1-4-5-6-10-12-1-17

Path 7 : 1-4-5-6-13-14-1-17

Path 8 : 1-4-5-6-13-15-1-17

Path 9 : 1-4-5-6-16-1-17

Jumlah jalur atau *path* yang didapat adalah 9. Nilai ini sesuai dengan perhitungan yang dilakukan pada kompleksitas sistematis.

4) Menentukan *Test Case*

Menurut IEEE Standart, *test case* adalah kumpulan dari input tes, kondisi yang akan dieksekusi, dan hasil yang diharapkan.

Test case berisi informasi tentang tujuan dari tes, kebutuhan perangkat lunak maupun perangkat keras, menspesifikasikan setup atau konfigurasi kebutuhan, gambaran mengenai bagaimana melakukan test tersebut, dan hasil yang diharapkan untuk test tersebut.

Test case yang akan dilakukan adalah dengan melakukan uji pada perangkat lunak berdasarkan pada *independent path* yang sudah dibuat.

Tabel 24. Uji *Test Case*

No	Nama	Hasil yang diharapkan	Hasil yang didapat	Ketercapaian
1	<i>Test Case 1</i>	Keluar dari aplikasi dan menampilkan kotak konfirmasi untuk keluar	Muncul kotak konfirmasi untuk keluar	Tercapai
2	<i>Test Case 2</i>	Menampilkan menu <i>Help</i> dengan sempurna	Menampilkan menu <i>Help</i> dengan baik	Tercapai
3	<i>Test Case 3</i>	Menampilkan fitur level satu (penggolongan 2 jenis sampah) dengan benar dengan kondisi menang	Fitur berjalan baik dan mampu menampilkan tampilan menang	Tercapai
4	<i>Test Case 4</i>	Menampilkan fitur level satu (penggolongan 2 jenis sampah) dengan benar dengan kondisi kalah	Fitur berjalan baik dan mampu menampilkan tampilan kalah	Tercapai

No	Nama	Hasil yang diharapkan	Hasil yang didapat	Ketercapaian
5	Test Case 5	Menampilkan fitur level dua (penggolongan 3 jenis sampah) dengan benar dengan kondisi menang	Fitur berjalan baik dan mampu menampilkan tampilan menang	Tercapai
6	Test Case 6	Menampilkan fitur level dua (penggolongan 3 jenis sampah) dengan benar dengan kondisi kalah	Fitur berjalan baik dan mampu menampilkan tampilan kalah	Tercapai
7	Test Case 7	Menampilkan fitur level dua (penggolongan 4 jenis sampah) dengan benar dengan kondisi menang	Fitur berjalan baik dan mampu menampilkan tampilan menang	Tercapai
8	Test Case 8	Menampilkan fitur level dua (penggolongan 4 jenis sampah) dengan benar dengan kondisi kalah	Fitur berjalan baik dan mampu menampilkan tampilan kalah	Tercapai
9	Test Case 9	Level-level yang ditampilkan dapat di reset seperti semula. Level 2 dan 3 terkunci.	Level kembali seperti pengaturan awal	Tercapai

Berdasarkan hasil pengujian pada tabel 24, dapat diketahui presentase ketercapaian sebagai berikut :

$$\text{Tercapai} = \frac{9}{9} \times 100\% = 100\%$$

$$\text{Gagal} = \frac{0}{9} \times 100\% = 0\%$$

Setelah didapatkan nilai kuantitatif kemudian dikonversikan menjadi nilai kualitatif berdasarkan skala penilaian media. Maka hasil pengujian *Whitebox* yang didapatkan adalah dengan interpretasi "**Sangat Baik**".

2. *Integration Testing*

Integration Testing merupakan tahap kedua pengujian perangkat lunak, dimana pengujian dilakukan menggunakan metode *blackbox* pada 3 *developer* untuk menguji aspek *fungsi* perangkat lunak.

a. Pengujian *Blackbox*

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *blackbox* didasarkan pada *usecase* yang telah dibuat.

Tabel 25. Hasil *Blackbox Test*

Nama Pengujian	Tujuan	Skenario	Hasil yang diharapkan	Penguji		
				1	2	3
Membuka Aplikasi	Mengetahui aplikasi dapat berjalan tanpa <i>error</i>	Mencoba membuka aplikasi	Aplikasi dapat berjalan dengan baik, menampilkan <i>splash screen</i> dan menuju menu	OK	OK	OK
Membuka <i>Play</i>	Agar pemain dapat memulai ke permainan	Membuka <i>Play</i> , menggunakan tombol <i>skip</i> untuk melewati intro	Menampilkan cerita intro dan menuju Menu memilih level	OK	OK	OK

Memilih Level	Agar pemain dapat memilih level yang diinginkan	Memilih level yang sudah terbuka dan belum terbuka	Hanya bisa memilih level yang terbuka	OK	OK	OK
Reset Level	Agar pemain dapat mengatur level seperti belum pernah dimainkan	Menggunakan tombol reset level	Data nilai menjadi 0, hanya level 1 yang terbuka	OK	OK	OK
Memainkan Game	Pemain memainkan permainan sesuai aturan	Memainkan <i>game</i> untuk dimenangkan dan dikalahkan	<i>Item</i> sampah muncul secara <i>random</i> , penambahan skor jika benar dan pengurangan skor jika salah	OK	OK	OK
Data Skor	Agar aplikasi mampu menyimpan data skor pada <i>database</i> di akhir permainan	Memenangkan permainan agar mampu membuka level selanjutnya	Data mampu tersimpan dan mampu membuka level selanjutnya	OK	OK	OK
Keluar Aplikasi	Agar pengguna dapat keluar dari aplikasi dengan benar	Menekan tombol keluar pada perangkat	Aplikasi dapat menutup dengan sempurna	OK	OK	OK

Berdasarkan hasil pengujian pada tabel 25, dapat diketahui presentase ketercapaian sebagai berikut :

$$\text{Tercapai} = \frac{21}{21} \times 100\% = 100\%$$

$$\text{Gagal} = \frac{0}{21} \times 100\% = 0\%$$

Setelah didapatkan nilai kuantitatif kemudian dikonversikan menjadi nilai kualitatif berdasarkan skala penilaian media. Maka hasil pengujian *Blackbox* menunjukkan interpretasi "**Sangat Baik**".

3. **System Testing**

System Testing merupakan tahap ketiga pengujian perangkat lunak, dimana pengujian dilakukan menggunakan metode *stress testing* untuk menguji aspek *reability* dan *instability test* untuk menguji aspek *compatibility*. Pengujian dilakukan dengan menggunakan bantuan *website testing tools* yang dapat diakses melalui <https://testobject.com/>.

a. **Install and Launch Testing**

Merupakan pengujian perangkat lunak dengan melakukan instalasi dan dijalankan pada berbagai macam perangkat. Pengujian dilakukan pada 6 macam perangkat yang berbeda-beda berdasarkan sistem operasi dan ukuran layar. Pengujian dilakukan dengan *testing tools* dan pada perangkat sesungguhnya. Berikut ini merupakan hasil pengujian aspek *compatibility* :

Tabel 26. Hasil pengujian *Compatibility*

Faktor Uji	Hasil uji dengan tool	Hasil uji pada perangkat
Sistem Operasi		
Android ICS 4.0.4	<i>Success</i>	<i>Success</i>
Android Jelly Bean 4.3.0	<i>Success</i>	<i>Success</i>
Android Kitkat 4.4.4	<i>Success</i>	<i>Success</i>
Resolusi		
480 x 800	<i>Success</i>	<i>Success</i>
1080 x 1920	<i>Success</i>	<i>Success</i>
1600 x 2560	<i>Success</i>	<i>Success</i>

Selain pengujian diatas peneliti juga melihat data dari *Playstore* terkait jumlah perangkat yang kompatibel untuk menjalankan *Game* Edunvi. Berdasarkan *database* pada *Playstore* *Game* Edunvi kompatibel pada 7993 perangkat Android.

The screenshot displays the Google Play Store page for the app 'EDUNVI : EDUKASI LINGKUNGAN'. At the top, the app name is followed by the package name 'com.wafda.edunvi' and a link to 'Lihat di Google Play Store'. Below this, there are three main sections: 'APK' with three options: 'PRODUKSI' (Version 10), 'PENGUJIAN BETA' (Siapkan uji coba Beta untuk aplikasi Anda), and 'PENGUJIAN ALPHA' (Siapkan uji coba Alfa untuk aplikasi Anda). Below these is a 'KONFIGURASI PRODUKSI' section with a button 'Unggah APK baru ke Produksi'. Further down, it states 'APK AKTIF' and 'dipublikasikan tanggal 14 Okt 2014 08.48.57'. At the bottom, there are two boxes: 'Perangkat yang didukung' showing '7993' with a link 'Lihat daftar', and 'Perangkat yang dikecualikan' showing '0' with a link 'Kelola perangkat yang dikecualikan'.

Gambar 34. Jumlah perangkat yang mendukung Edunvi menurut *Playstore*

Hasil pengujian kompatibilitas (lihat tabel 26) menunjukkan bahwa ketika *Game* Edunvi *diinstall* pada 3 versi sistem operasi android yang berbeda yaitu *ICS*, *Jellybean* dan *Kitkat* mampu berjalan dengan sempurna. *Game* Edunvi juga diterapkan pada 3 perangkat dengan ukuran resolusi layar yang berbeda yaitu 480x800, 1080 x 1920, dan 1600 x 2560 mampu berjalan dengan baik.

Berdasarkan hasil tersebut dapat disimpulkan bahwa presentase kesuksesan dari pengujian *compatibility* dengan menggunakan *Install and Lunch testing* adalah sebesar 100%

b. *Stress Testing*

Merupakan serangkaian peristiwa input (*touch*) pengguna virtual yang dilakukan secara acak dan berkali-kali untuk menguji aplikasi dibawah tekanan. Berikut adalah hasil *stress testing* yang dilakukan pada 6 macam perangkat yang berbeda :

Tabel 27. Hasil *Stress Testing*

Nama Perangkat	Hasil
Samsung Galaxy S Duos	<i>No bugs</i>
Sony Experia T	<i>No bugs</i>
Asus Google Nexus	<i>No bugs</i>
Nokia X	<i>No bugs</i>
LG Nexus 5	<i>No bugs</i>
Samsung Galaxy Tab 10	<i>No bugs</i>

Berdasarkan tabel 27 yang bersumber dari *quality report* (lihat lampiran) menunjukkan bahwa selama pengujian *stress, game* dapat berjalan dengan baik, tidak mengalami *error* dan tidak menunjukkan adanya penurunan performa yang mengganggu perangkat lunak sehingga dapat disimpulkan bahwa presentase kesuksesan dari pengujian *reability* dengan menggunakan *Stress Testing* adalah sebesar 100%.

4. *Acceptance Testing*

a. *Expert Judgement*

Expert judgement merupakan pengujian yang dilakukan dengan meminta pertimbangan atau pendapat dari para ahli atau orang yang berpengalaman dalam bidang tertentu. Menurut Azizah Jaafar(2010), pengujian ahli untuk Game Edukasi meliputi 5 Aspek yaitu *interface*, *pedagogical*, *content*, *multimedia* dan *playability*. Pengujian dilakukan setidaknya pada tiga orang ahli.

Berdasarkan hal tersebut, peneliti melakukan uji terhadap 3 orang ahli yang sudah mencakup kelima aspek tersebut. Pengujian dilakukan menggunakan kuesioner terhadap 3 orang ahli yaitu : Ahli Media, Ahli Materi dan Ahli Game. Hasil dari pengujian tersebut adalah sebagai berikut :

Tabel 28. Hasil Pengujian Ahli

Nama	Ponco Wali,M.Pd		
Keahlian	Media Pembelajaran		
Aspek Uji	<i>Multimedia, Interface</i>		
Jumlah Skor	Skor Maksimal	Presentase	Keterangan
42	48	87%	Baik
Nama	Dr. Insih Wilujeng, M.Pd		
Keahlian	Pendidikan IPA		
Aspek Uji	<i>Content, Pedagogical</i>		
Jumlah Skor	Skor Maksimal	Presentase	Keterangan
46	52	88%	Baik
Nama	Rahadian Pradipta		
Keahlian	Game Tester pada <i>Gameloft</i>		
Aspek Uji	<i>Playability</i>		
Jumlah Skor	Skor Maksimal	Presentase	Keterangan
19	19	100%	Sangat Baik

Pada tabel 28 , jumlah skor merepresentasikan total nilai dari yang diberikan oleh ahli dari keseluruhan pertanyaan, skor maksimal merepresentasikan total nilai maksimal yang dapat diberikan, presentase merepresentasikan presentase penilaian terhadap *Game* Edunvi oleh ahli. Hasil penilaian oleh ahli menunjukkan bahwa game Edunvi “Baik” dan layak untuk dijadikan penelitian.

Selain itu, para ahli juga memberikan saran untuk perbaikan *Game* Edunvi, antara lain :

Tabel 29. Saran dari Ahli

Aspek	Saran
<i>Multimedia</i>	Gambar sampah diperbaiki agar lebih familiar, ditambah tombol <i>replay</i> , ditambah sampah jenis lain.
<i>Pedagogical</i>	Aplikasi ini fungsinya lebih kepada penugasan/pengayaan guna melengkapi pembelajaran IPA
<i>Game</i>	Tombol <i>Back</i> pada permainan sebaiknya ke menu memilih level, bukan di halaman awal. Terdapat <i>bug</i> pada pilihan level.

Masukan dan saran yang diberikan oleh ahli nantinya akan dijadikan acuan untuk melakukan perbaikan terhadap *Game* Edunvi.

b. *Playability Testing*

Pengujian *playability* dilakukan terhadap pengguna Game Edunvi dari latar belakang pengguna aktif *Smartphone* Android sejumlah 20 responden. Pengujian ini dilakukan dengan membagikan kuesioner *Playability Heuristic for Mobile Game* oleh Hannu Korhonen (2006) yang

telah distandarkan. Kuisisioner ini terdiri dari 29 butir pertanyaan dengan skala 1-5. Hasil pengujian *playability* sebagai berikut :

Hasil nilai dari setiap responden kemudian dijumlahkan dan dibagi dengan skor maksimal yaitu 2900. Skor maksimal diperoleh dari skor terbesar yaitu 5 dikalikan dengan 29 butir pertanyaan dikalikan dengan 20 responden.

Tabel 30. Hasil Uji *Playability*

Responden	Total skor
1	111
2	120
3	102
4	109
5	125
6	98
7	113
8	106
9	92
10	113
11	118
12	125
13	130
14	125
15	126
16	129
17	132
18	136
19	133
20	138
Total Skor	2320
Skor Maksimum	2900
Presentase	82%

Berdasarkan tabel 30 dapat disimpulkan bahwa pada pengujian *playability* memperoleh presentase kelayakan sebesar 82% yang masuk dalam kategori **Baik**.

Sedangkan untuk menganalisis reabilitas dari hasil pengujian *playability* menggunakan metode *alpha cronbach* dengan bantuan *software* statistika SPSS dan mendapatkan hasil sebagai berikut :

Case Processing Summary			
		N	%
Cases	Valid	20	69.0
	Excluded ^a	9	31.0
	Total	29	100.0

a. Listwise deletion based on all variables in the procedure.

Reliability Statistics	
Cronbach's Alpha	N of Items
.923	29

Gambar 35. Hasil Penghitungan *Reability* dengan *Alpha Cronbach*

Dari hasil analisis *alpha cronbach* didapatkan konsistensi sebesar 0.923 sehingga apabila dicocokkan dengan indikator *Alpha Cronbach* dimana apabila nilai konsistensi lebih besar sama dengan 0.6 maka hasil pengujian dinilai realibel, sehingga dapat disimpulkan hasil pengujian *playability* dinyatakan realibel karena nilai konsistensi alpha 0.923 yang memiliki interpretasi **Tinggi**.

Hasil-hasil pengujian yang telah dilakukan, kemudian digunakan untuk mengetahui ketercapaian perangkat lunak terhadap kualitas perangkat lunak dari aspek *functionality*, *reliability*, *compatibility*, *playability*. Ketercapaian kualitas perangkat lunak tersebut ditunjukkan oleh tabel berikut :

Tabel 31. Hasil pengujian Game Edunvi

No	Aspek	Tahap Pengujian	Presentase	Tingkat Kelayakan
1	<i>Functionality</i>	<i>Unit Testing</i>	100%	Sangat baik
2	<i>Functionality</i>	<i>Integration Testing</i>	100%	Sangat baik
3	<i>Reliability</i>	<i>System Testing</i>	100%	Sangat baik
4	<i>Compatibility</i>	<i>System Testing</i>	100%	Sangat baik
5	<i>Playability</i>	<i>Acceptance Testing</i>	82%	Sangat Layak

Selanjutnya hasil dari rata-rata perhitungan pengujian aspek *functionality*, *reliability*, *compatibility* dan *playability* kemudian diubah data kualitatif menggunakan analisis Likert.

$$\text{Rata-rata presentase} = \frac{\text{jumlah nilai}}{\text{banyak nilai}} \times 100\%$$

$$\text{Rata-rata presentase} = \frac{480}{500} \times 100\% = 96\%$$

Setelah didapatkan hasil presentase rata-rata dari aspek pengujian tersebut sebesar 96% kemudian diubah ke pernyataan predikat yang digunakan dalam skala likert. Dari hasil tersebut dapat disimpulkan kualitas perangkat lunak *Game* Edunvi memiliki kategori "**Sangat Layak**"

C. Revisi Produk

Setelah melakukan berbagai pengujian, peneliti melakukan berbagai perbaikan terhadap aplikasi permainan *Game* Edunvi. Adapun perbaikan yang dilakukan yaitu :

Tabel 32. Daftar Perbaikan *Game* Edunvi

No	Deskripsi Perbaikan
1	Penggunaan tombol <i>exit</i> pada permainan menuju ke halaman memilih level
2	Perbaikan <i>bug</i> pada level 1. <i>Bug</i> terjadi ketika pemain memainkan level 1 sebanyak 2 kali maka level ke 3 akan terbuka
3	Perbaikan tampilan pada item sampah tas plastik

D. Hasil Akhir Produk

Aplikasi Permainan Edunvi ini telah dikembangkan sesuai dengan pedoman *System Development Life Cycle*. Tahapan-tahapan yang sudah dilalui yaitu analisis kebutuhan, analisis spesifikasi, desain, dan implementasi. Setelah perangkat lunak jadi, peneliti melakukan pengujian dalam rangka melakukan proses verifikasi dan validasi. Adapun proses verifikasi dengan melakukan pengujian unit dengan menggunakan metode *whitebox* dan pengujian integrasi dengan menggunakan metode *blackbox*. Sedangkan proses validasi yaitu dengan melakukan pengujian sistem dengan menggunakan metode *stress test* dan *instability test*, kemudian untuk pengujian *acceptance test* dengan menggunakan metode *expert review* dan *playability test*.

Aplikasi Game Edunvi dapat didownload secara gratis di *Playstore* dengan judul Edunvi : Edukasi Lingkungan. Hasil publikasi di *Playstore* adalah sebagai berikut :

Tabel 33. Hasil Publikasi *Game* Edunvi

Versi Edunvi	Waktu Publish	Jumlah <i>Download</i>
Edunvi 1.0	6 Maret 2014	54
Edunvi 1.3.5	22 Maret 2014	277
Edunvi 1.3.6	14 Oktober 2014	49

BAB V

SIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dalam mengembangkan *Game* Edunvi berbasis Android, maka dapat diambil kesimpulan sebagai berikut :

1. Pengembangan *Game* Edunvi menggunakan tahap-tahap pada V-model, yaitu analisis kebutuhan, analisis spesifikasi, desain, implementasi, pengujian unit, pengujian integrasi, pengujian sistem, dan uji penerimaan. V-model cocok untuk pengembangan *Game* Edunvi karena model ini dokumentasinya jelas untuk aplikasi yang tidak terlalu besar dan pengujian yang sesuai dengan tahap pengembangan.
2. Nilai presentase aspek *functionality* sebesar 100% dengan kategori "**sangat baik**". Pengujian aspek *reability* sebesar 100% dengan kategori "**sangat baik**", Pengujian aspek *compability* sebesar 100% dengan kategori "**sangat baik**". Pengujian aspek *playability* sebesar 82% dengan kategori "**sangat layak**" dan nilai *Alpha-Cronbach* sebesar 0.923
3. Berdasarkan hasil rata-rata pengujian aplikasi *Game* Edunvi kemudian didapatkan hasil presentase sebesar 96%, maka dari hasil tersebut *Game* Edunvi dapat memenuhi standar kualitas perangkat lunak karena memenuhi kategori kelayakan "**Sangat Layak**"

B. Saran

Penelitian ini masih terdapat banyak kekurangan dan hal-hal yang perlu dikaji dan dikembangkan lebih jauh lagi. Peneliti memiliki saran dan pemikiran untuk pengembangan kedepan antara lain :

1. Penambahan level baru pada permainan pada *Game* Edunvi.
2. Penambahan cara bermain yang bertipe *endless* agar dapat menyimpan *highscore* yang nantinya terintegrasi dengan akun pengguna pada *playstore*. Sehingga antar pengguna bias saling bersaing dalam mendapatkan highscore.
3. Pengembangan *Game* Edunvi pada *platform* lain dengan sistem operasi yang berbeda seperti IOS, Windows Phone, dll.
4. Penelitian lebih lanjut tentang aspek pengujian *Playability* untuk berbagai *platform game*.

DAFTAR PUSTAKA

- Adi Nugroho.(2010). *Rekayasa Perangkat Lunak Menggunakan UML dan JAVA*. Yogyakarta: Andi
- Alan F. (2014). *Jelly Bean still remains on 60.8% of Android devices; KitKat up to 8.5%*. http://www.phonearena.com/news/Jelly-Bean-still-remains-on-60.8-of-Android-devices-KitKat-up-to-8.5_id55777. Diakses pada 5 Januari 2015
- Appbrain (2015). *Most popular Google Play categories*. <http://www.appbrain.com/stats/android-market-app-categories>. Diakses pada 25 Februari 2015.
- Arief S. Sadiman. (2010). *Media pendidikan: pengertian, pengembangan dan pemanfaatannya*. Jakarta: Rajawali Pers
- Clark, Donald. (2006). *Games and e-learning* : www.caspianlearning.co.uk/Whhttpcaspian-games_1.1.pdf. Diakses tanggal 23 November 2014.
- Daryanto (2014). *Perlunya Pendidikan Lingkungan Hidup di Sekolah*. <http://www.vedcmalang.com/pppptkboemlg/index.php/menuutama/edukasi/996-perlunya-pendidikan-lingkungan-hidup-di-sekolah>. Diakses pada 25 Februari 2015
- Fowler, Martin.(2004). *UML Ditilled Edisi 3*. Yogyakarta: Andi
- Handriyantini. (2009). *Permainan Edukatif (Educational Games) Berbasis Komputer untuk Siswa Sekolah Dasar*. Malang: Sekolah Tinggi Informasi & Komputer Indonesia.
- Hasiah Mohamed & Azizah Jaafar. (2010).*Development and Potential Analysis of Heuristic Evaluation fo Educational Computer Game*. IEEE Computer Science Press.
- Hasiah Mohamed & Azizah Jaafar. (2010).*Heuristic Evaluation in Computer Game*. IEEE Computer Science Press.
- Heather Desurvire, Martin Caplan & Jozesef A. Toth. (2004). *Using Heuristics to Evaluate the Playability of Games*. IEEE Computer Science Press.
- Hannu Korhonen & Elina Koivisto. (2006). *Heuristics for Mobile Game*. IEEE Computer Science Press.
- Hannu Korhonen, Janne Paavilainen & Hannamari saarenpaa. (2009). *Expert Review Method in Game Evaluation*. IEEE Computer Science Press.

- Henderi. (2008). UML: Konsep dan Penerapannya Menggunakan Visual Paradigm. <http://www.blogster.com/henderi/uml-konsep-dan-penerapannya-menggunakan-visual-paradigm-171108195848>. Diakses pada 18 Februari 2015.
- J.F. DiMarzio (2008). *Android a Programmer Guide*. McGraw Hill Professional
- Kantor Utusan Khusus Presiden RI untuk MDGs. (2012). <http://mdgsindonesia.org/official/index.php/component/content/article/19-tulisan/artikel-lingkungan/37-manajemen-sampah>. Diakses tanggal 12 Februari 2015
- Kementrian Lingkungan Hidup. (2012). Waspadai Tumpukan Sampah di Sungai. <http://www.ampl.or.id/digilib/read/waspadai-tumpukan-sampah-di-sungai/34005>. Diakses tanggal 12 Februari 2015
- Mayer, R.E. (2009). *Multimedia Learning : Prinsip-prinsip dan Aplikasi*. Terjemahan oleh Teguh Wahyu. Yogyakarta: Pustaka Belajar.
- Mccabe, Thomas J. (1976). *A Complexity Measure*. IEE Transactions on Software Engineering.
- Myers, Glen (2004). *The Art of Software Testing, Second Edition*. New York: Wiley
- Nazarudin Safaat Harahap. (2012). *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika
- Nielsen, Jacob (2012). *How Many Test Users in a Usability Study?*. <http://www.nngroup.com/articles/how-many-test-users/>. Diakses pada tanggal 14 November 2014.
- Nielsen, Jacob (2012). *Quantitative Studies: How Many Users to Test?*. <http://www.nngroup.com/articles/quantitative-studies-how-many-users/>.
- Pressman, Roger S. (2010). *Software Engineering A Practitioner's Approach, Seventh Edition*. New York: McGraw Hill. Diakses pada tanggal 14 November 2014.
- Rosa, Shalahudin. (2013). *Rekayasa Perangkat Lunak Terstruktur dan berorientasi Objek*. Bandung: Informatika
- Sommerville, Ian.(2003). *Software Engineering*. Penerjemah: Dra. Yuhliza Hanum, M.Eng. Jakarta : Erlangga
- Sugiyono. (2003). *Metode Penelitian Administrasi dilengkapi dengan Metode R&D*. Bandung : Alfabeta.
- Sugiyono.(2008). *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta.

Teknojurnal. (2011). Smartphone : Semakin merakyat, semakin banyak dan semakin bergengsi. <http://teknojurnal.com/smartphone-semakin-merakyat-semakin-banyak-dan-semakin-bergengsi/>. Diakses pada tanggal 12 Februari 2015

Watkins, John.(2004). *Testing IT*. Cambridge: Cambridge University Press.

LAMPIRAN

Lampiran 1. Screenshot Event Edunvi

Global number	RndOb = 0
Global number	Miss = 0
Global number	Suhu = 25
Global number	Score = 0
Global number	VardbLama = 0
Global number	VardbScore = 0
Global number	wkt = 0

1 MENU			
2	Touch	On touched	but3
			Browser Close
			Audio Play dor not looping at volume 50 dB (tag "")
			Add action
3	Touch	On touched	but2
			System Go to help
			Audio Play dor not looping at volume 50 dB (tag "")
			Add action
4	Touch	On touched	but1
			System Go to intro
			Audio Play dor not looping at volume 50 dB (tag "")
			Add action

1 Pilih Level			
2	Touch	On touched	lv1
			Audio Play dor not looping at volume 50 dB (tag "")
			Set Visible
			Text Set text to "Buang sampah yang jatuh pada tempatnya : Organik & non Organik"
			Set Visible
			Set Visible
			Wait 5 seconds
			Go to level1
			Add action
3	Touch	On touched	lv2
	System		VardbScore > 0
			Audio Play dor not looping at volume 50 dB (tag "")
			Set Visible
			Text Set text to "Buang sampah yang jatuh pada tempatnya : Organik, non Organik & Plastik"
			Set Visible
			Set Visible
			Wait 5 seconds
			Go to level2
			Add action
4	Touch	On touched	lv3
	System		VardbScore > 1000
			Audio Play dor not looping at volume 50 dB (tag "")
			Set Visible
			Text Set text to "Buang sampah yang jatuh pada tempatnya : Kertas, Plastik, Logam & Kaca"
			Set Visible
			Set Visible
			Wait 5 seconds
			Go to level3
			Add action

1	Add Score		
2	→ ⚙️ System	On start of layout	⚙️ Reset global variables to default 🔊 Play bg_01 looping at volume 50 dB (tag "musik") ⚙️ Set Vardblama to <code>WebStorage.LocalValue("dbScore")</code> Add action
3	→ 🎮 Sprite14	On collision with 🗑️ Sprite11	⚙️ Add 100 to Score 📄 Set text to "score : " & Score 🗑️ Destroy 🔊 Play cling_01 not looping at volume 50 dB (tag "") Add action
4	→ 🎮 Sprite37	On collision with 🗑️ Sprite11	⚙️ Add 100 to Score 📄 Set text to "score : " & Score 🗑️ Destroy 🔊 Play cling_01 not looping at volume 50 dB (tag "") Add action
5	→ 🎮 s12	On collision with 🗑️ Sprite11	⚙️ Add 100 to Score 📄 Set text to "score : " & Score 🗑️ Destroy 🔊 Play cling_01 not looping at volume 50 dB (tag "") Add action
6	→ 🎮 Sprite26	On collision with 🗑️ Sprite11	⚙️ Add 100 to Score 📄 Set text to "score : " & Score 🗑️ Destroy 🔊 Play cling_01 not looping at volume 50 dB (tag "") Add action

12	Miss Score		
13	🎮 Sprite9	Y > 650	⚙️ Add 1 to Miss 📄 Set text to "MISS : " & miss 🗑️ Destroy Add action
14	→ 🎮 Sprite9	On collision with 🗑️ Sprite11	⚙️ Add 1 to Miss 📄 Set text to "MISS : " & miss 🗑️ Destroy Add action
15	🗑️ Sprite26	Y > 650	⚙️ Add 1 to Miss 📄 Set text to "MISS : " & miss 🗑️ Destroy Add action
16	→ 🗑️ Sprite26	On collision with 🗑️ Sprite8	⚙️ Add 1 to Miss 📄 Set text to "MISS : " & miss 🗑️ Destroy Add action
17	🗑️ Sprite14	Y > 650	⚙️ Add 1 to Miss 📄 Set text to "MISS : " & miss 🗑️ Destroy Add action

31	Moving Item		
	Pergerakan obj		
32	System	Every 0.2 seconds	<div> <div>Move 5 pixels at angle 90</div> <div>Rotate 15 degrees clockwise</div> <div>Move 5 pixels at angle 90</div> <div>Rotate 5 degrees clockwise</div> <div>Move 5 pixels at angle 90</div> <div>Rotate 20 degrees counter-clockwise</div> <div>Move 5 pixels at angle 90</div> <div>Rotate 5 degrees clockwise</div> <div>Move 5 pixels at angle 90</div> <div>Rotate 10 degrees counter-clockwise</div> <div>Move 5 pixels at angle 90</div> <div>Rotate 30 degrees clockwise</div> <div>Move 5 pixels at angle 90</div> <div>Rotate 40 degrees counter-clockwise</div> <div>Move 10 pixels at angle 90</div> <div>Rotate 50 degrees clockwise</div> <div>Move 10 pixels at angle 90</div> <div>Rotate 20 degrees counter-clockwise</div> <div>Set layer 0 opacity to 95</div> </div>

34	Random Item		
	random objek		
35	System	Every 5 seconds	<div> <div>Set RndOb to random(90)</div> <div>Set text to "randoman" & RndOb</div> </div>
36	System	$0 \leq \text{RndOb} \leq 10$	<div> <div>Create object Sprite5 on layer 0 at (random(200,1110), random(30))</div> </div>
37	System	$10 \leq \text{RndOb} \leq 20$	<div> <div>Create object Sprite9 on layer 0 at (random(200,1110), random(30))</div> </div>
38	System	$20 \leq \text{RndOb} \leq 30$	<div> <div>Create object Sprite14 on layer 0 at (random(200,1110), random(30))</div> </div>
39	System	$30 \leq \text{RndOb} \leq 40$	<div> <div>Create object Sprite2 on layer 0 at (random(200,1110), random(30))</div> </div>
40	System	$40 \leq \text{RndOb} \leq 50$	<div> <div>Create object Sprite on layer 0 at (random(200,1110), random(30))</div> </div>
41	System	$50 \leq \text{RndOb} \leq 60$	<div> <div>Create object s12 on layer 0 at (random(200,1110), random(30))</div> </div>
42	System	$60 \leq \text{RndOb} \leq 70$	<div> <div>Create object Sprite31 on layer 0 at (random(200,1110), random(30))</div> </div>
43	System	$70 \leq \text{RndOb} \leq 80$	<div> <div>Create object Sprite26 on layer 0 at (random(200,1110), random(30))</div> </div>
44	System	$80 \leq \text{RndOb} \leq 90$	<div> <div>Create object Sprite37 on layer 0 at (random(200,1110), random(30))</div> </div>

45 **Action Miss**

46	System	Miss = 0	<ul style="list-style-type: none"> Set Visible Set Invisible Set Invisible Set Invisible Set text to <i>suhu</i> & " 'C'"
47	System	Miss = 1	<ul style="list-style-type: none"> Set Visible Set Invisible Set Invisible Set Invisible Create object Sprite22 on layer 2 at (<i>random(200,1110)</i> , <i>random(710,740)</i>) Set text to <i>suhu</i> & " 'C'" Play SPLAT not looping at volume 50 dB (tag "")
48	System	Miss = 2	<ul style="list-style-type: none"> Set Invisible Set Visible Set Invisible Set Invisible Create object Sprite23 on layer 2 at (<i>random(200,1110)</i> , <i>random(710,740)</i>) Add 3 to Suhu Set text to <i>suhu</i> & " 'C'" Set Visible Play SPLAT not looping at volume 50 dB (tag "")

55 **Action Score**

56	System	Score = 300	<ul style="list-style-type: none"> Set Visible Set opacity to 50
57	System	Score = 500	<ul style="list-style-type: none"> Set Visible Set opacity to 100
58	System	Score = 600	<ul style="list-style-type: none"> Create object Sprite10 on layer 2 at (<i>random(200,1110)</i> , <i>random(580,650)</i>)
59	System	Score = 700	<ul style="list-style-type: none"> Create object Sprite16 on layer 2 at (<i>random(200,1110)</i> , <i>random(580,650)</i>)
60	System	Score = 800	<ul style="list-style-type: none"> Create object Sprite10 on layer 2 at (<i>random(200,1110)</i> , <i>random(580,650)</i>) Create object Sprite17 on layer 2 at (<i>random(200,1110)</i> , <i>random(580,650)</i>)
61	System	Score = 900	<ul style="list-style-type: none"> Set local key "dbScore" to 1000 Set VardbScore to 1000 Create object Sprite17 on layer 2 at (<i>random(200,1110)</i> , <i>random(580,650)</i>) Create object Sprite16 on layer 2 at (<i>random(200,1110)</i> , <i>random(580,650)</i>)
62			
63	System	Score = 1000	<ul style="list-style-type: none"> Stop "musik" Play tada not looping at volume 50 dB (tag "win")

Lampiran 2. *Quality Report* Edunvi

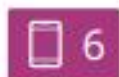


Quality Report

January 05, 2015 — 7:07 am



TESTS
were run.



DEVICES
were tested.



ERRORS
were automatically
detected.



BUGS
were reported.



Name from APK:
EDUNVI

Version from APK:
1.3.6

Your version name:
1.3.6

Package name:
com.wafda.edunvi

Uploaded:
January 05, 2015 — 5:04 am

Test Cases Explained

TestObject's Quality Report summarizes the results of automated test executions. The different test cases are listed below.

For details and FAQ, please visit our help section:
<https://help.testobject.com/docs/testing-tools/quality-report>

Install & Launch

The app is installed on various phones.

The app is installed and launched on all selected devices. In case the app does not install and launch properly, additional information is displayed.

Stress Test

A series of random user input events is sent to test the app under stress.

Thousands of user actions are simulated on your app. TestObject uses the Google Monkey Exerciser to execute random clicks within your app and provides information if an exception is thrown.

Screenshots

Screenshots of the app are taken on various phones.

Several screenshots of your app are taken on each of the selected devices. We automatically click through the app and take screenshots of several activities. You can review the screenshots and decide if the layout of your application looks alright or needs improvements.

Install & Launch



0 bugs were reported.

<div>▣ Asus Google Nexus 7 4.4.4</div> <div>Android 4.4.4 Screen: 800 x 1280 7"</div> <div>SUCCESS</div>	<div>▣ LG Nexus 5</div> <div>Android 5.0 Screen: 1080 x 1920 4.95"</div> <div>SUCCESS</div>
<div>▣ Nokia X</div> <div>Android 4.1.2 Screen: 480 x 800 4"</div> <div>SUCCESS</div>	<div>▣ Samsung Galaxy S Duos S7562</div> <div>Android 4.0.4 Screen: 480 x 800 4"</div> <div>SUCCESS</div>
<div>▣ Samsung Galaxy Tab Pro 10.1</div> <div>Android 4.4.2 Screen: 1600 x 2560 10.1"</div> <div>SUCCESS</div>	<div>▣ Sony Xperia T</div> <div>Android 4.3.0 Screen: 720 x 1280 4.55"</div> <div>SUCCESS</div>

Stress Test



bugs were reported.

<div>▣ Nokia X</div> <div>Android 4.1.2 Screen: 480 x 800 4"</div> <div>SUCCESS</div>	<div>▣ Samsung Galaxy Tab Pro 10.1</div> <div>Android 4.4.2 Screen: 1600 x 2560 10.1"</div> <div>SUCCESS</div>
--	---

Screenshots



0 bugs were reported.

▣ **Asus Google Nexus 7 4.4.4**

Android 4.4.4
Screen: 800 x 1280 | 7"

▣ **LG Nexus 5**

Android 5.0
Screen: 1080 x 1920 | 4.95"

▣ **Nokia X**

Android 4.1.2
Screen: 480 x 800 | 4"

▣ **Samsung Galaxy S Duos S7562**

Android 4.0.4
Screen: 480 x 800 | 4"

▣ **Samsung Galaxy Tab Pro 10.1**

Android 4.4.2
Screen: 1600 x 2560 | 10.1"

▣ **Sony Xperia T**

Android 4.3.0
Screen: 720 x 1280 | 4.55"

Asus Google Nexus 7 4.4.4

- Android 4.4.4
Screen: 800 x 800 | 7"



SUCCESS



SUCCESS



SUCCESS



SUCCESS



RunTime | January 26, 2015 10:20 AM
For details and FAQ, please visit our help section:
<http://help.testobject.com/testcloud-testing-tool-quality-report>

6/18



SUCCESS



SUCCESS



SUCCESS



SUCCESS

Samsung Galaxy S Duos S7562

ii Android 4.0.4
Screen: 480 x 480 | 4"



SUCCESS



SUCCESS



SUCCESS

Samsung Galaxy Tab Pro 10.1

Android 4.4.2

Screen: 1600 x 1600 | 10.1"



SUCCESS



SUCCESS



SUCCESS



EDUNVY | January 05, 2015 10:20 AM
For details and FAQ, please visit our help section:
<http://help.testobject.com/locutesting-tools/quality-report>

13718

Sony Xperia T

- Android 4.3.0
- Screen: 720 x 720 | 4.55"



SUCCESS



SUCCESS



SUCCESS

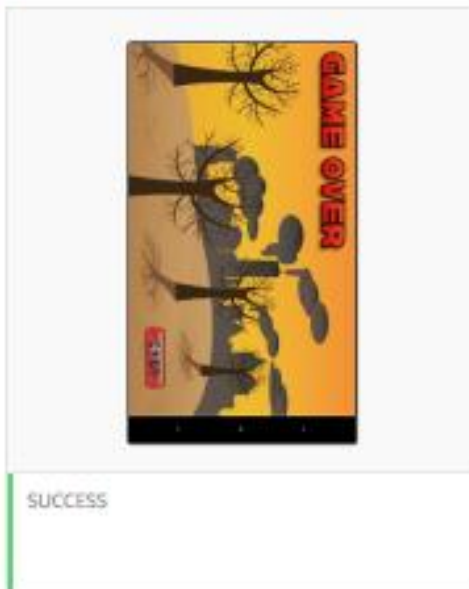
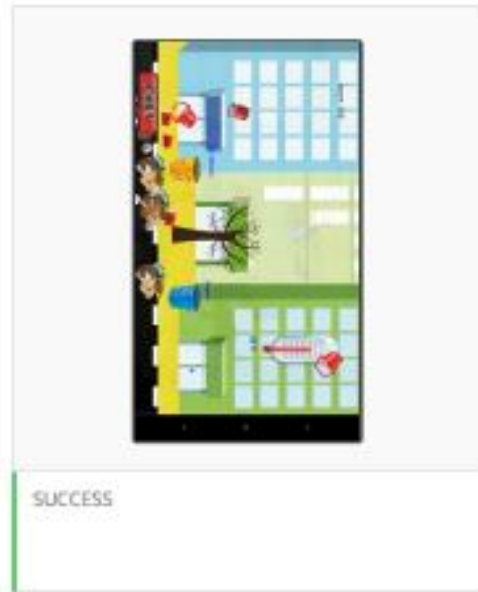
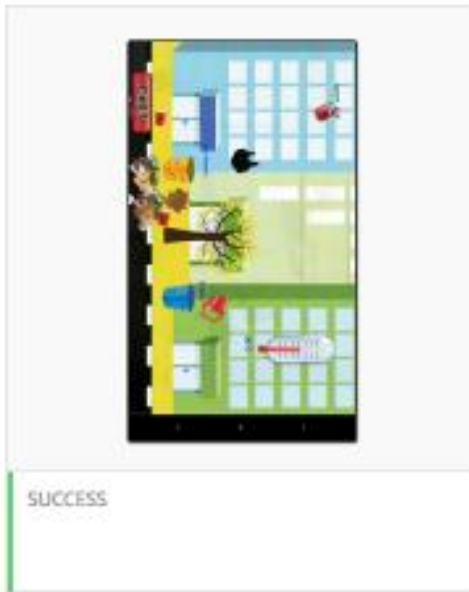


SUCCESS



62min | January 05, 2015 10:20 AM
For details and FAQ, please visit our help section:
<http://help.testobject.com/tester-testing-tools/quality-report>

13/14



Lampiran 3. Edunvi Pada Perangkat Android

Sistem Operasi Android	Tampilan pada Perangkat
Android ICS 4.0.4	
Android Jelly Bean 4.3.0	
Android Kitkat 4.4.4	

Resolusi Layar	
480 x 800	
1080 x 1920	
1600 x 2560	

Lampiran 4. Hasil Pengujian *Playability*

Responden	Pernyataan																												jumlah		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28		29	
1	4	4	4	4	4	4	4	4	4	3	3	4	4	3	4	4	4	4	3	4	4	5	3	4	3	3	4	4	4	5	111
2	4	3	4	3	4	4	4	5	5	4	4	5	4	3	5	4	5	5	5	4	4	5	3	3	3	4	5	4	4	5	120
3	4	4	4	3	3	4	4	4	3	5	4	2	2	4	4	4	4	3	4	4	3	4	3	2	3	2	4	4	4	4	102
4	4	3	5	2	4	5	4	5	3	2	4	4	5	5	2	4	3	2	4	4	5	4	3	4	3	4	4	3	5	109	
5	3	4	4	4	5	4	4	5	5	4	4	4	5	5	5	4	4	3	4	5	5	5	4	4	4	4	5	5	4	125	
6	4	4	4	4	4	4	3	2	3	2	4	4	3	4	3	4	4	4	4	4	2	3	4	2	2	3	3	4	3	98	
7	5	5	4	5	3	5	5	4	5	4	4	4	3	4	4	4	3	3	3	4	3	4	3	3	3	3	4	4	5	113	
8	5	5	5	5	5	4	5	4	4	3	4	5	4	3	3	4	4	4	2	3	3	3	2	2	3	3	3	2	4	106	
9	4	5	4	4	4	3	3	4	4	3	3	4	2	3	3	4	2	2	2	3	2	3	3	3	3	2	3	3	4	92	
10	5	5	5	5	5	5	5	5	5	5	4	4	3	4	3	3	3	3	3	4	3	3	3	3	2	4	3	4	4	113	
11	5	4	5	5	4	5	5	5	4	4	4	5	4	3	4	4	4	4	3	5	4	5	4	3	3	3	3	3	4	118	
12	5	4	5	5	5	5	5	5	5	5	5	5	4	3	3	4	4	3	4	5	3	4	3	4	4	4	4	5	5	125	
13	5	5	5	5	5	5	5	5	5	4	5	5	5	4	4	5	4	3	4	5	3	5	4	3	4	4	4	5	5	130	
14	5	4	5	5	4	4	5	5	5	5	5	5	4	4	4	5	3	3	4	4	4	5	3	4	4	3	5	5	4	125	
15	5	5	5	5	4	5	5	5	5	5	4	5	5	4	3	4	5	4	4	4	4	5	3	4	4	4	4	3	4	126	
16	5	5	5	5	5	5	5	5	5	5	5	5	3	3	3	4	5	5	5	5	4	4	4	3	4	4	5	4	4	129	
17	5	4	5	3	4	5	5	5	4	4	4	4	5	4	5	5	5	4	5	5	5	5	5	4	4	5	4	5	5	132	
18	5	5	5	5	5	5	5	5	5	5	4	5	5	4	4	5	5	5	5	4	4	4	4	5	4	5	5	4	5	136	
19	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	5	4	4	4	4	4	5	4	4	4	5	5	4	4	133	
20	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4	5	5	5	4	4	4	5	5	4	5	138	

Lampiran 5. Validasi Instrumen

SURAT PERNYATAAN VALIDASI INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Dessy Irmawati, M.T
NIP : 19791214 201012 2 002
Jurusan : Pendidikan Teknik Elektronika

Menyatakan bahwa instrumen penelitian TAS atas nama mahasiswa:

Nama : Wafda Adita Rifai
NIM : 10520241036
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Analisis Pengembangan Game Edunvi berbasis Android

Setelah dilakukan kajian atas instrumen penelitian TAS tersebut dapat dinyatakan:

- ☒ Layak digunakan untuk penelitian
☐ Layak digunakan dengan perbaikan
☐ Tidak layak digunakan untuk penelitian yang bersangkutan

dengan saran/perbaikan sebagaimana terlampir.

Demikian agar dapat digunakan sebagaimana mestinya.

Yogyakarta,

Validator,

Dessy Irmawati, MT

NIP. 19791214 201012 2 002

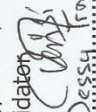
Catatan :

☐ Beri tanda ✓

Hasil Validasi Instrumen Penelitian TAS

Nama Mahasiswa : Wafda Adita Rifai
Judul TAS : Analisis Pengembangan Game Edunvi berbasis Android

No	Variabel	Saran/Tanggapan
	Komentar Umum/Lain-lain: Instrumen sudah layak digunakan.	

Yogyakarta,
Validasi

.....Dessy Perawati MT.....
NIP. 19791214 20102 2 002

**SURAT PERNYATAAN VALIDASI
INSTRUMEN PENELITIAN TUGAS AKHIR SKRIPSI**

Saya yang bertanda tangan dibawah ini :

Nama : Drs. Suparman M Pd
NIP : 19491231 197803 1 009
Jurusan : Pendidikan Teknik Elektronika

Menyatakan bahwa instrumen penelitian TAS atas nama mahasiswa:

Nama : Wafda Adita Rifai
NIM : 10520241036
Program Studi : Pendidikan Teknik Informatika
Judul TAS : Analisis Pengembangan Game Edunvi berbasis Android

Setelah dilakukan kajian atas instrumen penelitian TAS tersebut dapat dinyatakan:

- ☐ Layak digunakan untuk penelitian
☒ Layak digunakan dengan perbaikan
☐ Tidak layak digunakan untuk penelitian yang bersangkutan
dengan saran/perbaikan sebagaimana terlampir.

Demikian agar dapat digunakan sebagaimana mestinya.

Yogyakarta,

Validator,

Drs. Suparman M Pd

NIP. 19491231 197803 1 009

Catatan :

☐ Beri tanda ✓

Hasil Validasi Instrumen Penelitian TAS

Nama Mahasiswa : Wafda Adita Rifai
 Judul TAS : Analisis Pengembangan Game Edukasi berbasis Android

No	Variabel	Saran/Tanggapan
	Kreativitas	Dg kata/kalimat yg lb tepat/snei
	Komentar Umum/Lain-lain:	

Yogyakarta,
 Validator,



.....
 NIP. 19991231 197803009

Lampiran 6. Hasil *Expert Judgement*

Lembar Evaluasi Aplikasi Permainan Edunvi Oleh Ahli Media

Materi : Lingkungan
Sasaran : Anak-anak umur 8-14 tahun
Judul Penelitian : Analisis dan Pengembangan Game Edunvi berbasis Android
Peneliti : Wafda Adita Rifai
Evaluator :
Pekerjaan / Jabatan :

Deskripsi :

Lembar evaluasi ini digunakan untuk menilai kualitas media dari aplikasi permainan Edunvi. Aplikasi ini digunakan sebagai media pendukung pembelajaran tentang kepedulian lingkungan melalui pengelolaan sampah. Sehubungan dengan hal tersebut, Bapak / Ibu sebagai ahli media dimohon untuk memberikan tanggapan dan komentar / saran terhadap media pembelajaran ini.

Petunjuk :

1. Lembar evaluasi ini diisi oleh Ahli Media.
2. Pada rentangan tanggapan terdapat 4 (empat) tingkatan.
3. Berilah tanda cek [✓] pada kolom sesuai dengan pendapat anda sesuai dengan keadaan yang sebenarnya.
4. Jawaban diberikan pada kolom skala penilaian yang sudah disediakan dengan skala penilaian :
1 = STS (Sangat Tidak Setuju)
2 = TS (Tidak Setuju)
3 = S (Setuju)
4 = SS (Sangat Setuju)
5. Terimakasih atas kesediaan Bapak / Ibu untuk mengisi lembar evaluasi ini.

No	Kriteria Penilaian	Tanggapan			
		1	2	3	4
Antarmuka					
1.	Tampilan konsisten dan sesuai standar perangkat lunak				✓
2	Antar aktivitas pada Game Edunvi dapat dimengerti oleh semua kalangan				✓
3	Game Edunvi menyajikan petunjuk yang jelas			✓	
4	Navigasi antarmuka disajikan sesuai aturan yang berlaku				✓
5	Tampilan disusun secara baik				✓
6	Penggunaan warna, font, teks dapat dibaca dengan mudah				✓
7	Kualitas antarmuka dapat diterima dan enak dilihat				✓
Multimedia					
8	Penggunaan elemen multimedia (teks, animasi, gambar, suara dan video) sesuai dengan konteks				✓
9	Kombinasi antar elemen multimedia dapat dimengerti				✓
10	Penyajian multimedia diatur dengan baik				✓
11	Ketepatan penggunaan konten multimedia dalam menyampaikan informasi			✓	
12	Tidak berlebihan dalam menggunakan elemen multimedia dalam satu layar				✓

Komentar / saran umum :

- perlu ditambah tombol reply
- gambar sampah tas plastik di edit lebih familiar
- ditambah gambar kain / baju bekas.

Kesimpulan :

Aplikasi permainan Edunvi berbasis android dinyatakan :

- ☐ Dapat digunakan tanpa perbaikan
- ☒ Dapat digunakan dengan perbaikan
- ☐ Tidak dapat digunakan

Yogyakarta, 18-12-2014

Ahli Media



ponco walip.....

NIP.

**Lembar Evaluasi
Aplikasi Permainan Edunvi
Oleh Ahli Aplikasi Permainan**

Materi : Lingkungan
Sasaran : Anak-anak umur 8-14 tahun
Judul Penelitian : Analisis dan Pengembangan Game Edunvi berbasis Android
Peneliti : Wafda Adita Rifai
Evaluator :
Pekerjaan / Jabatan :

Deskripsi :

Lembar evaluasi ini digunakan untuk menilai kualitas dari aplikasi permainan Edunvi. Aplikasi ini digunakan sebagai media pendukung pembelajaran tentang kepedulian lingkungan melalui pengelolaan sampah. Sehubungan dengan hal tersebut, Bapak / Ibu sebagai ahli media dimohon untuk memberikan tanggapan dan komentar / saran terhadap media pembelajaran ini.

Petunjuk :

1. Lembar evaluasi ini diisi oleh Ahli Aplikasi Permainan
2. Berilah tanda cek [✓] pada kolom sesuai dengan pendapat anda sesuai dengan keadaan yang sebenarnya.
3. Jawaban diberikan pada kolom taraf ketercapaian yang sudah disediakan dengan skala penilaian : Ya atau tidak
4. Terimakasih atas kesediaan Bapak / Ibu untuk mengisi lembar evaluasi ini.

No	Aktifasi/ Menu	Hasil yang diharapkan	Taraf ketercapaian	
			Ya	Tidak
1	Membuka aplikasi	Menampilkan halaman menu Edunvi yang terdiri menu Play dan Help	✓	
2	Membuka menu Help	Menampilkan macam-macam <i>item</i> sampah	✓	
		Menampilkan deskripsi sampah ketika item sampah disentuh	✓	
3	Membuka menu Play	Menampilkan cerita/intro permainan	✓	
		Menampilkan menu pilih level terdiri dari 3 macam level yang terbuka ketika sudah menyelesaikan level sebelumnya	✓	
4	Reset data skor	Menampilkan pesan peringatan data akan dihapus	✓	
		Level yang sudah terbuka menjadi terkunci kembali	✓	
5	Memilih salah satu level	Menampilkan petunjuk permainan selama sesi <i>loading</i>	✓	
6	Memainkan game	Menampilkan sampah secara acak	✓	
		Menampilkan tempat sampah sesuai level	✓	
		Sampah dapat di <i>drag and drop</i>	✓	
		Skor bertambah jika benar	✓	
		Memunculkan sampah berserakan jika salah dalam membuang sampah	✓	
		Daun pada pohon akan gugur jika selalu salah	✓	
		Nilai termometer akan naik jika selalu salah	✓	
		Menampilkan rumput dan bunga jika benar dalam membuang sampah	✓	
		Menampilkan halaman menang jika berhasil	✓	
		Menampilkan halaman <i>game over</i> jika gagal	✓	
7	Menutup Aplikasi	Menampilkan pesan peringatan jika akan keluar	✓	

Komentar / saran umum :

Terdapat beberapa bug yang major, antara lain :

1. Pada saat memilih opsi "back", game akan kembali ke halaman awal, bukan menu memilih game. (pada posisi memainkan level).
2. Jika level 1 dimainkan dua kali akan membuka level 3, tanpa melalui/menyelesaikan level 2.


Kesimpulan :

Aplikasi permainan Edunvi berbasis android dinyatakan :

- ☐ Dapat digunakan tanpa perbaikan
- ☒ Dapat digunakan dengan perbaikan
- ☐ Tidak dapat digunakan

Yogyakarta, November 2014

Ahli Aplikasi Permainan



(RAHADIAN WAHYU.P)

NIP.

**Lembar Evaluasi
Aplikasi Permainan Edunvi
Oleh Ahli Pendidikan**

Materi : Lingkungan
Sasaran : Anak-anak umur 8-14 tahun
Judul Penelitian : Analisis dan Pengembangan Game Edunvi berbasis Android
Peneliti : Wafda Adita Rifai
Evaluator :
Pekerjaan / Jabatan :

Deskripsi :

Lembar evaluasi ini digunakan untuk menilai kualitas materi dari aplikasi permainan Edunvi. Aplikasi ini digunakan sebagai media pendukung pembelajaran tentang kepedulian lingkungan melalui pengelolaan sampah. Sehubungan dengan hal tersebut, Bapak / Ibu sebagai ahli media dimohon untuk memberikan tanggapan dan komentar / saran terhadap media pembelajaran ini.

Petunjuk :

1. Lembar evaluasi ini diisi oleh Ahli Pendidikan.
2. Pada rentangan tanggapan terdapat 4 (empat) tingkatan.
3. Berilah tanda cek [✓] pada kolom sesuai dengan pendapat anda sesuai dengan keadaan yang sebenarnya.
4. Jawaban diberikan pada kolom skala penilaian yang sudah disediakan dengan skala penilaian :
1 = STS (Sangat Tidak Setuju)
2 = TS (Tidak Setuju)
3 = S (Setuju)
4 = SS (Sangat Setuju)
5. Lembar evaluasi ini disertai lampiran berupa silabus.
6. Terimakasih atas kesediaan Bapak / Ibu untuk mengisi lembar evaluasi ini.

No	Kriteria Penilaian	Tanggapan			
		1	2	3	4
Kualitas Materi					
1	Game Edunvi ini lengkap dan dapat digunakan untuk alat bantu pembelajaran <i>materi penggolongan sampah</i>			✓	
2	Materi penggolongan sampah sesuai dengan kaidah jenis-jenis sampah				✓
3	Penggunaan contoh-contoh sampah dalam permainan sering ditemui dalam kehidupan sehari-hari				✓
4	Ilustrasi yang digambarkan tentang efek membuang sampah sesuai realita yang sesungguhnya			✓	
5	Materi disampaikan secara bertahap			✓	
6	Game Edunvi memberikan umpan balik/ <i>feedback</i> secara langsung terhadap tindakan pengguna.				✓
7	Materi yang disampaikan mudah diterima dan dipahami oleh anak			✓	
8	Game Edunvi meningkatkan pemahaman anak dan lebih jelas dalam membedakan sampah berdasarkan jenisnya				✓
9	Game Edunvi dapat memberikan pengetahuan baru bagi siswa				✓
10	Penyampaian materi menggunakan model permainan drag and drop untuk membiasakan anak dalam menerapkan di kehidupan sehari-hari			✓	
Kemanfaatan					
11	Penggunaan game Edunvi ini dapat membantu guru/orangtua dalam mengajarkan kepedulian lingkungan				✓
12	Penggunaan game Edunvi menarik minat anak dalam belajar tentang kepedulian lingkungan				✓
13	Penggunaan game Edunvi melatih anak dalam belajar secara mandiri			✓	

Komentar / saran umum :

Aplikasi permainan ini, fungsinya lebih
kepada penugasan / pengayaan guna
melengkapi pemb. IPA. (perlu ada penjelasan)

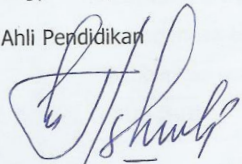
Kesimpulan :

Aplikasi permainan Edunvi berbasis android dinyatakan :

- ☒ Dapat digunakan tanpa perbaikan
- ☐ Dapat digunakan dengan perbaikan
- ☐ Tidak dapat digunakan

Yogyakarta,

Ahli Pendidikan



Dr. Ihsan Wilyeng, M.Pd

NIP. 196712021993032001

KARTU BIMBINGAN SKRIPSI (Untuk Mahasiswa)

FRM/EKA/05-00
25 Januari 2008

Nama Mahasiswa : Wafelo Adita Rifau

No. Mahasiswa : 10520241036

E-mail :

Program Studi : 1. Pendidikan Teknik Elektronika Jenjang : S1

Pendidikan Teknik Informatika
Jenjang : S1

Kelas : E - 2010

Dosen Pembimbing : Dr. Eko Marpanaji

Judul : Analisis Pengembangan Game Edukasi Berbasis Android

No	Tanggal	Uraian Bimbingan	Tandatangan Pembimbing
1.	12/3	Bab 1. Latar Belakang	
2.	15/4	Teori Pengembangan	
3.	9/6	RPL	
4.	23/8	Pengujian perangkat lunak	
5.	9/10	Bab III Metode Penelitian	
6.	18/11	Bab IV Pembahasan	
7.	10/12	Hasil Penelitian	
8.	5/1	Analisis Data Pengujian	
9.	16/1	Bab V. Kesimpulan dan Saran	
10.			

Rekomendasi Pembimbing :

1. Mahasiswa yang bersangkutan siap untuk diuji.

Tanggal Persetujuan : 20-1-2015

2. Kartu Bimbingan ini wajib dilampirkan pada saat pendaftaran ujian Skripsi.

KARTU MONITORING SKRIPSI (Untuk Dosen Pembimbing)

FRM/EKA/06-00
25 Januari 2008

Nama Mahasiswa : Warda Adita Rofai

Nama Mahasiswa : Daffa Fauzan
No. Mahasiswa : 10520241036

NO. 17445152
E-mail : 10320241036

Program Studi : 1. Pendidikan Teknik Elektronika Jenjang : SI

2 Pendidikan Teknik Informatika

Kelas : E-2016

Dosen Pembimbing : Dr. Agus Mardiana, S.Pd, M.Pd

Judul : Analisis Pengembangan Game Edukatif berbasis Android

No	Tanggal	Catatan Pembimbing
1.		sebutkan penelitian yg relevan
2.		Nekronologi sejarah dg popul
3.		Desain & pengembangannya
4.		Software testing & pengujian
5.		testrunners & kriterium pengujian
6.		semua hasil pengujian di plaske
7.		testrunner & testcases pengujian
8.		pengulas gambar
9.		testrunner & testrunner
10.		

Keterangan :

Mahasiswa yang bersangkutan telah disetujui untuk ujian Skripsi.

Tanggall Persetujuan

Tandatangan

Lampiran 8. Surat Keterangan Pembimbing

**9KEPUTUSAN DEKAN
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
NOMOR : 258/ELK/Q-I/XI/2013
TENTANG
PENGANGKATAN, N PEMBIMBING TUGAS AKHIR SKRIPSI
BAGI MAHASISWA FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA**

**DEKAN FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA**

Menimbang : 1. Bahwa sehubungan dengan telah dipenuhi syarat untuk penulisan Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta, perlu diangkat pembimbing.
2. Bahwa untuk keperluan dimaksud perlu ditetapkan dengan Keputusan Dekan.

Mengingat : 1. Undang-undang Nomor 20 tahun 2003.
2. Peraturan Pemerintah RI Nomor 60 tahun 1999.
3. Keputusan Presiden RI: a. Nomor 93 tahun 1999; b. 305/M tahun 1999.
4. Keputusan Menteri Pendidikan dan Kebudayaan RI: Nomor 274/O/1999.
5. Keputusan Mendiknas RI Nomor 003/O/2001.
6. Keputusan Rektor UNY Nomor : 1160/UN34/KP/2011

MEMUTUSKAN

Menetapkan
Pertama : Mengangkat Pembimbing Tugas Akhir Skripsi bagi mahasiswa Fakultas Teknik Universitas Negeri Yogyakarta sebagai berikut :

Nama Pembimbing	: Dr. Eko Marpanaji
Bagi mahasiswa	:
Nama/No.Mahasiswa	: Wafda Adita Rifai / 10520241036
Jurusan/ Prodi	: Pendidikan Teknik Elektronika / Pendidikan Teknik Informatika
Judul Skripsi	: Analisis Pengembangan Game Edmu Berbasis Android

Kedua : Dosen pembimbing disertai tugas membimbing penulisan Tugas Akhir Skripsi sesuai dengan Pedoman Tugas Akhir Skripsi.

Ketiga : Keputusan ini berlaku sejak ditetapkan

Keempat : Segala sesuatu akan diubah dan dibetulkan sebagaimana mestinya apabila di kemudian hari ternyata terdapat kekeliruan dalam Keputusan ini.


Ditandatangani : di Yogyakarta
Pada tanggal : 14 November 2013
Dit. Moch. Bfuri Triyono
NIP: 19560216 198603 1 003

Tembusan Yth :
1. Wakil Dekan II FT UNY
2. Ketua Jurusan Pendidikan Teknik Elektronika
3. Kasub. Bag Pendidikan FT UNY
4. Yang bersangkutan